UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

The Compartmentalised Knapsack Problem:
a case study

Robinson Hoto;
Marcos Arenales
Nelson Maculan

Nº 81

NOTAS

USP
ICMC

São Carlos - SP

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2577

The Compartmentalised Knapsack Problem:
a case study

Robinson Hoto
Marcos Arenales
Nelson Maculan

Nº 81

NOTAS

Série Computação

São Carlos – SP
Out./2004

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2577

The Compartmentalised Knapsack Problem: a case study

Robinson Hoto
Marcos Arenales
Nelson Maculan

Nº 81

NOTAS

Série Computação

São Carlos – SP
Out./2004

Hoto, R, Arenales, M., Maculan, N. *The compartmentalised Knapsack Problem: A case Study*, Technical Report n. 81, ICMC-USP, 2004.

The Compartmentalised Knapsack Problem: a case study

Robinson Hoto [a] Marcos Arenales [b] Nelson Maculan [c]

[a] *Department of Mathematics, State University of Londrina - Londrina, PR, Brazil*
[b] *ICMC/Computer Science and Statistics Department, University of São Paulo - São Carlos, SP, Brazil*
[c] *COPPE/Systems Engineering, Federal University of Rio de Janeiro - Rio de Janeiro, RJ, Brazil*

Abstract

The Compartmentalised Knapsack Problem (CKP) can be stated as the following. A climber wants to load his/her knapsack with items of various classes (e.g., medicine, food, tools, clothes) and for each item an weight and a utility value are given. However, the climber cannot put items of different classes together, so he/she must arrange the available space inside the knapsack into compartments, and put only items of the same class into the same compartment. The capacity of each compartment is lower and upper bound, and there are costs and loss of knapsack capacity if a compartment is created. The climber has to decide how many compartments should be created and how to load them in order to maximise the total utility value. This paper addresses a case study in the cutting of steel rolls in which the CKP arises. The rolls are cut in two-phases: the first phase produces sub-rolls (compartments) which are, after reducing thickness, cut in a second phase to produce ribbons (a class consists of ordered items with the same thickness). Finally, two methods of solving CKP are presented, which are used to generate columns in the classical linear optmisation model of Gilmore and Gomory. Moreover, computational experiments are presented.

Keywords: Cutting and packing problem; Knapsack problems; Combinatorial optimisation.

Resumo

O Problema da Mochila Compartimentada (PMC) pode ser enunciado da seguinte forma: um alpinista quer carregar sua mochila com itens de diversas classes (p. ex., medicamentos, alimentos, ferramentas, roupas, etc.) e, para cada item é associado seu peso e um valor de utilidade. O alpinista não pode colocar juntos itens de classes diferentes, de modo que precisa organizar o espaço interno da mochila em compartimentos e colocar itens de mesma classe em compartimentos. A capacidade de cada compartimento é limitada superior e inferiormente e, além disso, há custos e perdas da capacidade da mochila para cada compartimento criado. O alpinista tem de decidir quantos compartimentos deveriam ser criados e como devem ser carregados de modo a maximizar o valor de utilidade, descontado os custos por compartimento criado. Este artigo está orientado para um estudo de caso de corte de bobinas de aço, que surge em indústrias metalúrgicas. Os rolos em estoque são cortados em duas fases: a primeira fase produz sub-rolos (compartimentos), os quais são recortados em fitas demandadas, depois de sofrerem um processo de laminação (redução de espessura. Uma classe consiste de itens demandados de mesma espessura). Finalmente, dois métodos de resolução do PMC são apresentados, os quais são usados para gerar colunas na clássica abordagem de Gilmore e Gomory e resultados computacionais são apresentados.

Palavras-chaves: Problema de corte e empacotamento, Problema da Mochila, Otimização Inteira e Combinatória

---

1. Introduction

Since Gilmore and Gomory's pioneer articles (1961, 1963) on 1-dimensional cutting stock problems, a number of paper have been published on this theme. This paper deals with a kind of 1-dimensional cutting stock problem that arises in steel industries. Clients order ribbons of different widths and thicknesses. The thicknesses of ribbons may match or not the thickness of the steel rolls in stock. The ribbons are grouped into classes of the same thickness. Therefore, the rolls must be cut in two phases. Firstly, a roll is cut into sub-rolls whereby their thicknesses are reduced to match the ribbon thickness (Fig.1). Then, in a second phase the reduced thickness sub-rolls are cut independently using the ribbon width of the same class (Fig. 2). There are some technical losses to produce sub-rolls and their widths are also upper and lower bound. One should cut the rolls considering such characteristics, so that waste should be minimised, and since the reducing thickness process is expensive, the number of sub-rolls should also minimised. This characterises a compartmentalised knapsack problem, as described in the abstract.

Gilmore and Gomory's linear optmisation cutting stock model (1965) is appropriated when consider this problem, however a cutting pattern that corresponds to a column of the input matrix has to be generated by considering the phases described above. In other words, each time a column is generated a compartmentalised knapsack problem has to be solved.

Only few articles in the literature are related to the compartmentalised knapsack problem, however they avoided dealing with it explicitly by reducing or relaxing the solution space. In the case of reducing the solution space, feasibility is maintained and in the case of relaxing the solution space, simple heuristics are used to exclude non-compartmentalised cuttings.

Carvalho and Rodrigues (1995) reduced the solution space in such a way that the sub-rolls are cut to produce only one type of item, i.e., a homogeneous cutting pattern (pattern 1 in Fig. 3). Cutting pattern 2 in Fig. 3 is not obtained by this reduced search.

Haesller (1979), Ferreira *et al.* (1990) and Correia *et al.* (2004) generated ordinary 1-dimensional cutting patterns without compartmentalisation constraints, and by using simple heuristics they tried to organise the items into compartments or discard the cutting pattern.

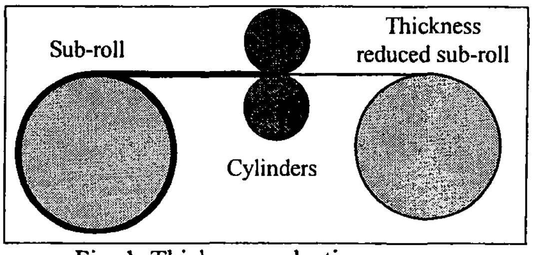

Fig. 1. Thickness reduction process

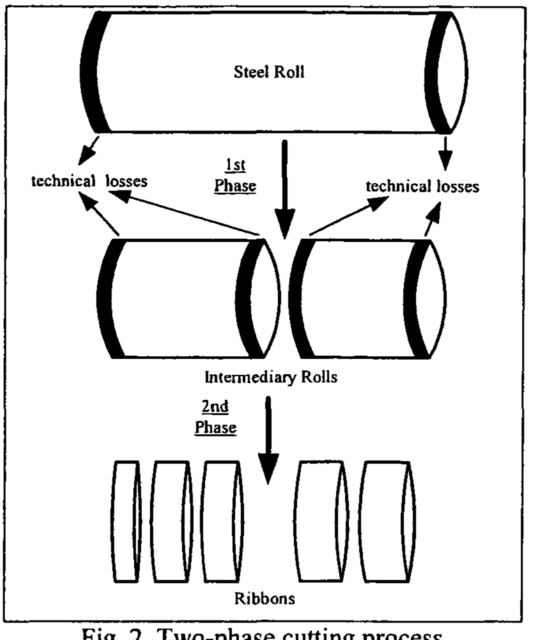

Fig. 2. Two-phase cutting process

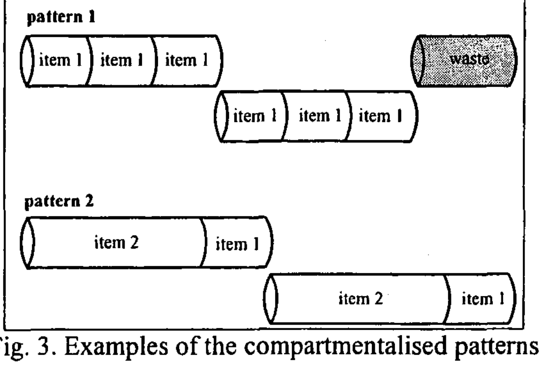

Fig. 3. Examples of the compartmentalised patterns

Zak (2002) studied a two-phase cutting problem and proposed a column-row generation method. In the row generation phase, a new intermediate roll is considered. Although the problem that Zak attacked can be seen as building compartments, it differs from what we consider as a compartmentalised knapsack problem since any item can be put together in any compartment. Regarding his problem, which arises in the paper industry, the two phases (or more) arise because of the cutting process and the items are not grouped into classes.

In section 2, we present the unconstrained compartmentalised knapsack problem and describe two solution methods. In section 3, the constrained compartmentalised knapsack is considered, which is necessary in the case of low demand. In section 4, the cutting stock problem to meet the demand is considered. In section 5, we present a simple rounding heuristic to obtain a feasible integer solution. Finally, the computational experiments are given in section 6.

2. The Unconstrained Compartmentalised Knapsack Problem

Consider a knapsack problem with the capacity given by c to be loaded with n items. For each item $i \in N = \{1,\dots,n\}$ two integer numbers are given: its utility value u_i and its weight p_i. The classical knapsack problem is to determine the number of items to be loaded in such a way that the knapsack capacity is not exceeded and the total utility value is maximised. Assume now that the items are grouped into classes (e.g., food, clothes, medicine, etc.) and compartments should be built inside the knapsack so that only items of the same class are loaded into a same compartment. This defines a partition $\{N_1,\dots,N_k\}$ on set N, that is, there are k classes of items. The capacity of each compartment to be loaded with items of class s has a lower limit, say $d_s^{\min}$, and an upper, say, $d_s^{\max}$, $s = 1,\dots,k$.

In order to prevent trivial infeasibility, it is assumed that: $p_i \le d_s^{\max} \le c$ for all $i \in N_s$, $s = 1,\dots,k$.

For $s = w_6 = 1,\dots,k$, consider the set of all integer non-negative linear combination of the weights: $w_h = \sum_{i \in N_s} a_i p_i$, $a_i \ge 0$ and integer, such that $d_s^{\min} \le w_h \le d_s^{\max}$, $h \in V_s$, where $V_1 \cup \cdots \cup V_k = V$ is the set of indexes of all combinations, $h = 1$, $h = 2$, etc, such that $\{V_1,\dots,V_k\}$ define a partition on set V. The capacities for compartments can be restricted to w_h without loss of generality.

For the sake of clarity we exemplify the sets with a simple instance. Let $N_1 = \{1, 2\}$, $N_2 = \{3, 4, 5\}$ (i.e., m=5 items of k=2 classes) and the weights are $p_1 = 3$, $p_2 = 4$, $p_3 = 3$, $p_4 = 6$, $p_5 = 7$. Also, let $d_1^{\min} = d_2^{\min} = 3$ and $d_1^{\max} = d_2^{\max} = 10$.

Then, the integer non-negative linear combinations of weights are given by: for $s = 1$, $w_1 = 3$, $w_2 = 4$, $w_3 = 6$, $w_4 = 7$, $w_5 = 8, 9$, $w_7 = 10$, then $V_1 = \{1, 2, 3, 4, 5, 6, 7\}$ and for $s = 2$, $w_8 = 3$, $w_9 = 6$, $w_{10} = 6$, $w_{11} = 7$, $w_{12} = 9$, $w_{13} = 9$, $w_{14} = 10$, then $V_2 = \{8, 9, 10, 11, 12, 13, 14\}$. Note that, with this notation, when we refer to $h = 6$ ($w_6 = 9$), subset N_1 is implicitly considered, although combination $w = 9$ can also be found in another set, for instance, $w_{12} = 9$, but when we refer to h=12, the subset N_2 is considered. Also, note that $w_{12} = 3p_3 + 0p_4 + 0p_5 = 9$ and $w_{13} = 1p_3 + 1p_4 + 0p_5 = 9$, i.e., there are two different ways of combining items of class 2 to obtain the width w=9. Next, we consider the width combinations without referring to the way of obtaining it.

Let W_s, $W_s \subseteq V_s$ such that only different values of the combinations are considered. For example, $W_2 = \{8, 9, 11, 12, 14\}$. In fact, *w-values* can be generated without knowing a_i, $i \in N_s$. See Christofides and Whitlock (1977) or Morabito and Arenales (1995) for a recursive formula. To make modelling easier we use V_s, but in order to devise methods we use W_s. To obtain the linear combination, solve an ordinary knapsack (see step 2 in the COMP_ex method).

Therefore, for each h, we refer to a particular compartment of width w_h, composed by items of class s, where $h \in V_s$, and for which a cost λ_h of creating compartment h is given. The cost θ_h is problem dependent (e.g., in steel roll cutting, it is a cost of reducing the sub-roll thickness to match the item thickness). Let a_{ih} be the number of items i in the compartment h and y_h be the number of times the compartment h is in the knapsack. The Compartmentalised Knapsack Problem (CKP) can be modeled as the following:

$$max\ z = \sum_{h \in V_1} \left(\left(\sum_{i \in N_1} u_i a_{ih} \right) - \lambda_h \right) y_h + \cdots + \sum_{h \in V_k} \left(\left(\sum_{i \in N_k} u_i a_{ih} \right) - \lambda_h \right) y_h \quad (1.1)$$

subject to:

$$\sum_{h \in V_1} \sum_{i \in N_1} p_i a_{ih} y_h + \cdots + \sum_{h \in V_k} \sum_{i \in N_k} p_i a_{ih} y_h \le c \quad (1.2)$$

$$y_h \ge 0 \text{ and integers, } h \in V \quad (1.3)$$

For the sake of simplicity, consider in model (1.1) – (1.3) $v_h = \sum_{i \in N_s} u_i a_{ih}$ (the utility value of the compartment $h \in V_s$) and $w_h = \sum_{i \in N_s} p_i a_{ih}$ (the capacity of the compartment $h \in V_s$).

Two methods were proposed for CKP (Hoto *et al.* (2002). The first one evaluates exactly the compartments and the other one heuristically.

The COMP_ex Method

Now we describe an exact method for solving Unconstrained Compartmentalised Knapsack Problems. The idea consists of generating all the best possible compartments.

1. For $s = 1, \ldots, k$, determine all possible non-negative linear combinations of p_i, $i \in N_s$, that is, $w_h = \sum_{i \in N_s} a_{ih} p_i$, $a_{ih} \ge 0$ and integer (for this use the recursive formula in Christofides and Whitlock, 1977, or Morabito and Arenales, 1995) such that $d_s^{\min} \le w_h \le d_s^{\max}$, and enumerate them as described before, which define set W_s.

2. For $s = 1, \ldots, k$ and all $h \in W_s$ solve the knapsack problem:

$$v_h = max \sum_{i \in N_s} u_i a_{ih}$$

subject to:

$$\sum_{i \in N_s} p_i a_{ih} = w_h, \ a_{ih} \ge 0 \text{ and integer, } i \in N_s$$

3. Solve the knapsack problem:

$$max \sum_{h \in W} (v_h - \lambda_h) y_h$$

subject to:

$$\sum_{h \in W} w_h y_h \le c, \ y_h \ge 0 \text{ and integer,}$$

$$\text{where } h \in W = W_1 \cup \cdots \cup W_k$$

A drawback of this algorithm is in step 2, where a large number of knapsack problems should be solved, since the total number of linear combinations in step 1 can be very large. In order to overcome this problem, we substitute v_h for an upper bound.

The COMP_bound Method

Here we describe a heuristic method, which consists of evaluating each compartment without solving it. We use the upper bound given by Martello and Toth (1990, p. 93).

1. For $s = 1, \ldots, k$, determine all possible non-negative linear combinations of p_i, $i \in N_s$, that is, $w_h = \sum_{i \in N_s} a_{ih} p_i$, where $a_{ih} \ge 0$ and integer, and such that $d_s^{\min} \le w_h \le d_s^{\max}$ and enumerate them as described before.

2. For $s = 1, \ldots, k$ and $h \in W_s$ determine an upper bound $\bar{v}_h$ to the knapsack problem in step 2 of COMP_ex method: $v_h \le \bar{v}_h$.

3. Solve the knapsack problem:

$$\max \sum_{h\in W} (\bar{v}_h - \lambda_h) y_h$$
subject to:
$$\sum_{h\in W} w_h y_h \le c, \; y_h \ge 0 \textit{ and integer}, \; h \in W$$

Remark: For compartments h such that $y_h > 0$ in step 3, solve knapsack problems in step 2 of COMP_ex to obtain the loadings. Of course, an iterative method could be designed as the following: after solving the knapsack problems to obtain the loadings, use v_h instead of $\bar{v}_h$ and repeat step 3. Repeat until no more changings. Experiments with this iterative method can be found in Hoto (2001).

3. The Constrained Compartmentalisation

The constrained CKP arises if demand is low, so that the repetition of an item in a cutting pattern has to be limited. That is, the number of items type i in the overall knapsack cannot exceed b_i.

Letting $v_h = \sum_{i\in N_s} u_i a_{ih}$ and $w_h = \sum_{i\in N_s} p_i a_{ih}$, the mathematical model for the constrained CKP can be given as follows:

$$\max z = \sum_{h\in V_1} (v_h - \lambda_h) y_h + \cdots + \sum_{h\in V_k} (v_h - \lambda_h) y_h \quad (2.1)$$

subject to:

$$\sum_{s=1}^{k} \sum_{h\in V_s} w_h y_h \le c \quad (2.2)$$

$$\sum_{s=1}^{k} \sum_{h\in V_s} a_{ih} y_h \le b_i, \; i \in N \quad (2.3)$$

$$y_h \ge 0 \textit{ and integers}, \; h \in V \quad (2.4)$$

Now we describe a greedy heuristic for solving the constrained CKP. The idea consists of generating compartments of widths varying from $d_s^{\min}$ to $d_s^{\max}$ for each class s, such that $a_{ih} \le \bar{b}_i$, where in the begining $\bar{b}_i = b_i$ and it is updated until $\bar{b}_i = 0$. Therefore, we generated a number of compartments for the same class which can compose the overall knapsack without exceeding the demand. A 0-1 knapsack problem is finally solved to choose the compartments.

COMPREST Method

1. For $s = 1, \ldots, k$, determine all possible non-negative linear combinations of p_i, $i \in N_s$, that is, $w_h = \sum_{i\in N_s} a_{ih} p_i$, $a_{ih} \ge 0$ and integer (for this use the recursive formula in Christofides and Whitlock, 1977, or Morabito and Arenales, 1995) such that $d_s^{\min} \le w_h \le d_s^{\max}$, and enumerate them as described before, which define the set W_s. Let $\bar{b}_i = b_i$, $i=1,\ldots,n$.

2. For $s = 1, \ldots, k$ and all $h \in W_s$ repeat until $\bar{b}_i = 0$, for all $i \in N_s$:

2.1 Solve the constrained knapsack problem:

$$v_h = \max \sum_{i\in N_s} u_i a_{ih}$$

subject to:

$$\sum_{i\in N_s} p_i a_{ih} = w_h$$

$$0 \le a_{ih} \le \bar{b}_i$$

$$a_{ih} \ge 0 \textit{ and integer}, \; i \in N_s$$

2.2 Update $\bar{b}_i \leftarrow \bar{b}_i - a_{ih}$, and solve the next knapsack.

3. Solve the 0-1 knapsack problem:

$$\max \sum_{h\in W} (v_h - \lambda_h) y_h$$

subject to:

$$\sum_{h\in W} w_h y_h \le c$$

$$y_h = 0 \text{ or } 1, \; h \in W$$

In the COMPREST, the 0-1 knapsack problem in step 3 prevents exceeding the demand (each compartment is only used once).

Similarly to the heuristic method COMP_bound, in step 2, we may use an upper bound $\bar{v}_h$ to the knapsack problem, and we call this method as COMPREST_bound.

Furthermore, if the number of knives is limited, it is necessary to limit the number of items in the compartments. Let f be the number of available knives to the first phase and g be the number of knives to the second one, then the following constraint has to be added to the knapsack problem in step 2:

$$\sum_{i \in N_s} a_{ih} \leq g-1 \leq g\text{-}1,$$

and the following constraint has to be added to the knapsack problem in step 3:

$$\sum_{h \in V} y_h \leq f-1,$$

(see Gilmore and Gomory, 1963).

4. The Compartmentalised Cutting Stock Problem

So far, we have considered cutting problems that consist of cutting just one roll. When demand is to be met, and a number of rolls has be cut, a mathematical model consists of determining how many rolls should be cut according to a set of given cutting patterns. The approach to be used is the one given by Gilmore and Gomory (1961, 1965). First of all, in order to write down the mathematical model, assume that all compartmentalised cutting patterns have already been generated (to solve the model, a column generation technique is used, i.e., a compartmentalised cutting pattern is generated in each iteration of the simplex method). The problem is better defined as follows:

Roll Data

Consider the indexes: $r = 1, \ldots, m$, where m is the number of types of rolls, and $j = 1, \ldots, p^r$, where p^r is the number of possible compartimentalised cutting patterns for the roll r. Truly, p^r is in practice a very large number and it is not necessary to find it when a column generation technique is used. Let,

- L^r width (mm) of roll r.
- P^r weight (kg) of the roll r,
- e^r available (units) of the roll r,
- δ^r unit linear cost ($/mm) of the roll r,
- T^{jr} waste (mm) in the roll r, if cutting pattern j is used,
- c^{jr} cost of cutting the roll r according to pattern j.

Ribbon Data (items)

Consider the index: $i = 1, \ldots, n$, where n is number of types of ordered ribbons. These items are grouped into k classes: $N_1, \ldots, N_k$.

- ℓ_i width (mm) of the ribbon i,
- b_i demand (kg) for the ribbon i.

Intermediate Rolls (compartments)

Consider the index: $h = 1, \ldots, H^r$, where H^r is the number of compartments for the roll type r (in the example of the section 2, $H^r = 14$) and let:

- y_h^{jr} the number of compartments type h in the cutting pattern j for the roll r,
- a_i^{jr} number of ribbons i in the cutting pattern j for the roll r,

  (note that $a_i^{jr} = \sum_{h=1}^{H^r} a_{ih}^{jr} y_h^{jr}$, where a_{ih}^{jr} is the number of ribbons type i in the compartment h of the j-th cutting pattern for the roll r. Remember that if $h \in V_s$ then the items in the compartment h belong to N_s. It should be clear that a_{ih}^{jr} and y_h^{jr} are generated, e.g., by the COMP_ex method. Here, for the purpose of writing the model, we consider all possible solutions for the compartmentalised knapsack problem),
- θ_h^r the cost of reducing the thickness of the roll r to the thickness of items in the compartment h.

Therefore, the cost c^{jr} may be given by:

$$c_j^r = \delta^r T^{jr} + \sum_{h=1}^{H^r} (\theta_h^r y_h^{jr}).$$

The decision variables

- x^{jr} is the number of the rolls type r cut according to the cutting pattern j.

A mathematical model for the cutting stock problem can then be stated as follows:

$$min \sum_{r=1}^{m}\sum_{j=1}^{p^r} c^{jr}x^{jr} \quad (3.1)$$

subject to:

$$\sum_{r=1}^{m}\sum_{j=1}^{p^r}\frac{P^r}{L^r}\ell_i a_i^{jr}x^{jr}=b_i,\ i=1,\dots,n \quad (3.2)$$

$$\sum_{j=1}^{p^1} x^{j1} \le e^1$$

$$\ddots \qquad \vdots \qquad (3.3)$$

$$\sum_{j=1}^{p^m} x^{jm} \le e^m$$

$$x^{jr}\ge 0,\ \textit{integer},\ r=1,\dots,m, j=1,\dots,p^r \quad (3.4)$$

Column Generation (ColGen)

Note that each column of the constraint matrix in (3.2) – (3.3) has $(n+m)$ elements and the column of the variable x^{jr} is given by:

$$\alpha^{jr}=\left(\alpha_1^{jr},\alpha_2^{jr},\dots,\alpha_n^{jr},0,\dots,0,1,0,\dots,0\right)^T,$$

where $\alpha_i^{jr}=\frac{P^r}{L^r}\ell_i a_i^{jr}$ and the number one is in the $(n+r)$-*th* position.

Assume that B is a feasible basis at a simplex iteration (an initial feasible basis made of simple compartmentalised cutting patterns is given below). The simplex multiplier is given by:

$$\pi = c_B B^{-1} =$$
$$\left(\pi_1,\dots,\pi_n,\pi_{n+1},\dots,\pi_{n+r},\dots,\pi_{n+m}\right).$$

In order to verify the optimality condition, we calculate the reduced cost of a variable x^{jr} which is given by:

$$c_j^r-\pi\alpha_j^r=\delta^r T^{jr}+\sum_{h=1}^{H_r}\theta_h^r y_h^{jr}-\sum_{i=1}^{n}\pi_i\alpha_i^{jr}-\pi_{n+r}$$

Note that, by considering S_1 and S_2 as technical losses,

$T^{jr}=(L^r-S_1-S_2\sum_{h=1}^{H_r}y_h^{jr})-\sum_{i=1}^{n}\ell_i a_i^{jr}$, then:

$$c_j^r-\pi\alpha_j^r=\delta^r\left((L^r-S_1-S_2\sum_{h=1}^{H_r}y_h^{jr})-\sum_{i=1}^{n}\ell_i a_i^{jr}\right)+$$
$$\sum_{h=1}^{H_r}\theta_h^r y_h^{jr}-\sum_{i=1}^{n}\pi_i\frac{P^r}{L^r}\ell_i a_i^{jr}-\pi_{n+r}=\delta^r(L^r-S_1)-$$
$$\delta^r S_2\sum_{h=1}^{H_r}y_h^{jr}-\delta^r\sum_{i=1}^{n}\ell_i a_i^{jr}+\sum_{h=1}^{H_r}\theta_h^r y_h^{jr}-$$
$$\sum_{i=1}^{n}\pi_i\frac{P^r}{L^r}\ell_i a_i^{jr}-\pi_{n+r}.$$

Therefore,

$$c_j^r-\pi\alpha_j^r=-\sum_{i=1}^{n}(\delta^r+\pi_i\frac{P^r}{L^r})\ell_i a_i^{jr}+$$
$$\sum_{h=1}^{H_r}(\theta_h^r-\delta^r S_2)y_h^{jr}+\delta^r(L^r-S_1)-\pi_{n+r}.$$

Remember that $a_i^{jr}=\sum_{h=1}^{H^r}a_{ih}^{jr}y_h^{jr}$, then:

$$c_j^r-\pi\alpha_j^r=-\sum_{i=1}^{n}\left((\delta^r+\pi_i\frac{P^r}{L^r})\ell_i\sum_{h=1}^{H^r}a_{ih}^{jr}y_h^{jr}\right)+$$
$$\sum_{h=1}^{H_r}(\theta_h^r+\delta^r S_2)y_h^{jr}+\delta^r(L^r-S_1)-\pi_{n+r}=$$
$$-\sum_{h=1}^{H^r}\sum_{i=1}^{n}\left([(\delta^r+\pi_i\frac{P^r}{L^r})\ell_i]a_{ih}^{jr}y_h^{jr}\right)+$$
$$\sum_{h=1}^{H_r}(\theta_h^r-\delta^r S_2)y_h^{jr}+\delta^r(L^r-S_1)-\pi_{n+r}.$$

Finally,

$$c_j^r-\pi\alpha_j^r=-\sum_{h=1}^{H^r}\left((\sum_{i=1}^{n}u_i a_{ih}^{jr})-\lambda_h\right)y_h^{jr}+$$
$$(\delta^r(L^r-S_1)-\pi_{n+r}) \quad (4)$$

where,

$$u_i=(\delta^r+\pi_i\frac{P^r}{L^r})\ell_i \text{ and } \lambda_h=\theta_h^r-\delta^r S_2.$$

Since the last term in (4) is independent of the cutting pattern, the minimum reduced cost can be determined by:

$$min\ \{-\sum_{h=1}^{H^r}\left((\sum_{i=1}^{n} u_i a_{ih}^{jr}) - \lambda_h\right) y_h^{jr},\ j=1,\ldots,p^r \ \ and\ \ r=1,\ldots,m\} \quad (5)$$

By using the notation in section 2, the set of all possible compartments for roll r is grouped into the compartments for each class, that is, $\{1,\ldots,H^r\} = V_1 \cup \cdots \cup V_k$, where V_s contains the indexes of compartments for class s, also $\{1,\ldots,n\} = N_1 \cup \cdots \cup N_k$. Therefore, the problem (5) consists of solving the compartmentalised knapsack problem, for every roll r (see problem (1.1) – (1.3)):

$$max\ z = \sum_{h\in V_1}\left((\sum_{i\in N_1} u_i a_{ih}) - \lambda_h\right) y_h + \cdots + \sum_{h\in V_k}\left((\sum_{i\in N_k} u_i a_{ih}) - \lambda_h\right) y_h \quad (6.1)$$

subject to:

$$S_1 + \sum_{h\in V_1}(S_2 + \sum_{i\in N_1}\ell_i a_{ih}) y_h + \cdots + \sum_{h\in V_k}(S_2 + \sum_{i\in N_k}\ell_i a_{ih}) y_h \le L^r \quad (6.2)$$

$$y_h \ge 0 \ \ and\ integers,\ h\in V \quad (6.3)$$

Note that in (6.2), the width of a compartment $h \in V_s$ (i.e., an intermediate roll), which is one of items in class N_s, has to include a technical loss S_2, then: $w_h = S_2 + \sum_{i\in N_s}\ell_i a_{ih}$. Also, the loss S_1 is included for the first stage of cutting.

Solving the Cutting Stock Problem

In order to solve the problem (3.1) – (3.4) we used the simplex method with the column generation technique for the relaxed problem, i.e., by relaxing the integer condition. Each column is generated by the solution of the problem (6.1) – (6.4) obtained by COMP_ex or COMP_bound presented in section 2. To obtain a feasible integer solution we used rounding heuristics described in the next section.

The initial basic matrix is built by homogeneous compartmentalised cutting patterns:

Choose the roll r such that $e^r P^r = \max\{e^t P^t,\ t=1,\ldots,m\}$;

for $i=1$ **to** n **do for** $j=1$ **to** n **do** $a_i^{jr} := 0$,

{building homogenous patterns}

for $i=1$ **to** n **do for** $j=1$ **to** n **do**

if $i=j$ **then begin**

$$a_i^{jr} := \left\lfloor \frac{L_{\max}}{\ell_i} \right\rfloor,$$

{width of the homogenous compartment}

$w := a_i^{jr}\ell_i$ *{in practice,* $w > L_{\min}$*}*

{total of compartments in the knapsack}

$$y := \left\lfloor \frac{L^r}{w} \right\rfloor,$$

{total of items in the knapsack}

$$a_i^{jr} := y a_i^{jr}$$

end

end;

The basis is completed with slack variables of (3.3). If the slack variable for stock r is negative, then, an artificial variable is used for this constraint and phase I of the simplex method is performed.

5. Rounding of the Solution

In this section we present a rounding procedure to obtain an integer solution, based on Wäscher and Gau (1996). Suppose that the linear relaxation of problem (3.1) – (3.4) has already been solved to obtain a continuous solution $\hat{x}$. Then, round down $\hat{x}$ to obtain an integer solution $\lfloor \hat{x} \rfloor$. Unless the continuous solution is already integer, the rounded solution will not meet demand, remaining a residual demand:

$$\hat{b}_i = b_i - \sum_{r=1}^{m}\sum_{j=1}^{p^r}\frac{P^r}{L^r}\ell_i a_{ij}^r \left\lfloor \hat{x}^{jr} \right\rfloor,\ i=1,\ldots,n.$$

Also, the availablility for the roll r is updated:

$$\hat{e}^r = e^r - \sum_{j=1}^{p^r}\left\lfloor \hat{x}^{jr} \right\rfloor,\ r=1,\ldots,m.$$

Therefore, a residual problem is formulated to meet the remaining demand, similar to the problem (3.1) – (3.4):

Residual Problem

$$min \sum_{r=1}^{m} \sum_{j=1}^{p^r} c^{jr} x^{jr} \qquad (7.1)$$

subject to:

$$\sum_{r=1}^{m} \sum_{j=1}^{p^r} \frac{P^r}{L^r} \ell_i a_i^{jr} x^{jr} = \hat{b}_i,\ i = 1, \ldots, n \qquad (7.2)$$

$$\begin{array}{lll} \sum_{j=1}^{p^1} x^{j1} & & \le \hat{e}^1 \\ & \ddots & \vdots \\ & & \sum_{j=1}^{p^m} x^{jm} \le \hat{e}^m \end{array} \qquad (7.3)$$

$$x^{jr} \ge 0,\ integer,\ r=1,\ldots,m, j=1,\ldots, p^r \qquad (7.4)$$

Note that the residual problems have a low demand. Therefore, to generate the cutting patterns to residual problems it is necessary to add restriction (8) to the problem (6.1) – (6.4), and solve the constrained CKP with COMPREST or COMPREST_bound.

$$\sum_{s=1}^{k} \sum_{h \in V_s} \frac{P^r}{L^r} \ell_i a_{ih} y_h \le \hat{b}_i,\ i = 1, \ldots, n \qquad (8)$$

After solving the last residual problem, it is still possible that a small demand lasts, for which we may use an exhausting repetition heuristic (Hinxman, 1980), that consists of generating a constrained compartmentalised pattern and use it as much as possible. It is repeated until there is no more demand. More elaborated rounding heuristics can be devised (see Stadtler (1990), Wäscher and Gau (1996)).

6. Computational Results

The methods presented in this paper were implemented by using Borland Delphi 7.0 and run on a Intel Celeron Processor 2.8 GHz with 512 MB of memory. Here we compare the perfomance of solving the cutting stock problem by generating columns in four different ways, since the overall method has two phases: Phase I is to solve the linear relaxation of model (3.1.) – (3.4), and Phase II is of rounding to obtain an integer solution. For Phase I the unconstrained compartmentalised knapsack problem is solved for each simplex iteration, but for Phase II, the constrained compartmentalised knapsack problem has to be solved to deal with low demand in the residual problems. The four ways are given by:

- **Combination 1**
  Phase I: COMP_bound
  Phase II: COMPREST_bound
- **Combination 2**
  Phase I: COMP_bound
  Phase II: COMPREST
- **Combination 3**
  Phase I: COMP_ex
  Phase II: COMPREST_bound
- **Combination 4**
  Phase I: COMP_ex
  Phase II: COMPREST

A number of instances were randomly generated based on typical sizes found in practice:

Item data:

- k (the number of classes): 5, 10, 15, 20, 30 and 40.
- n (the number of items types): randomly generated from 4 to 6 items for each class N_s.
- ℓ_i (width item): randomly generated so that, 55 mm $\le \ell_i \le$ 250 mm.
- b_i (demand): randomly generated so that, 87,000 kg $\le b_i \le$ 99,000 kg.

Roll data:

- r (number of roll types): 3.
- L^r (width of roll r): 900 mm, 1100 mm and 1200 mm.
- P^r (weigth of roll r): 12,000 kg, for all r.
- e^r (available of roll r).
- δ^r (unit linear cost of roll r): $\delta^r = 1$, for all r.

In the columns **class** and **items** are respectively the number of classes and the number of items of each instance. In the column **time** is the running time in minutes, seconds and milliseconds, and the column **interm** are the number of intermediate rolls generated for each combination. In the column **waste** is the percentage of waste, without the technical losses.

Note in Table 1, if the COMP_bound in the Phase I is used (i.e., combinations 1 and 2), then the method tends to build more intermediate rolls than if the COMP_ex is used (i.e., combinations 3 and 4).

The use of COMPREST_bound or COMPREST in Phase II is not significant for the number of intermediate rolls generated. On the other hand, if COMPREST is used in Phase II, the running time is greater.

7. Conclusions

In this paper we use the Compartmentalised Knapsack Problem (CKP) to generate the cutting patterns of the rolls of steel (CCSP), because the thicknesses of the rolls may not match the thicknesses of the ordered items, so the rolls have to be cut into intermediate rolls, whose thicknesses are reduced to match the thicknesses of the items.

We present a basic and exact method to the unconstrained CKP which was modified to a heuristic one. The ideas were extended to the constrained CKP.

In order to obtain a continuous solution of the Compartmentalised Cutting Stock Problem we used Gilmore and Gomory's column generation technique, and to obtain a feasible integer solution we used a simple rounding heuristic.

The preliminary computational results suggest that the combination of COMP_ex (unconstrained cutting patterns) with COMPREST_bound (constrained cutting patterns) result in a good strategy.

Acknowledgements

This work was supported by CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, and by FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo. The authors would like to thank Professor Nei Yoshihiro Soma for his useful comments and hints about knapsack implementations.

References

J. M. V. Valério de CARVALHO and A. J. G. RODRIGUES (1994). A computer based interactive approach to a two-stage cutting-stock problem. INFOR, 32(4), 243-252.

J. M. V. Valério de CARVALHO and A. J. G. RODRIGUES (1995). An LP-based approach to a two-stage cutting-stock problem. European Journal of Operational Research, 84, 580-589.

J. S. FERREIRA, M. A. NEVES and P. F. CASTRO (1990). A two-phase roll cutting problem. European Journal of Operational Research, 44, 185-196.

M. H. CORREIA, J. F. OLIVEIRA and J. S. FERREIRA (2004). Reel and Sheet Cutting at a Paper Mill, Computers and Operations Research, 31, 1223, 1243.

P. C. GILMORE and R. E. GOMORY (1961). A Linear Programming Approach to the Cutting Stock Problem, Operations Research, 9, 849-859.

P. C. GILMORE and R. E. GOMORY (1963). A Linear Programming Approach to the Cutting Stock Problem, parth II, Operations Research, 14, 94-120.

P. C. GILMORE and R. E. GOMORY (1966). The theory and computation of knapsack functions. Operations Research, 14, 1045-1074.

R. W. HAESSLER (1975). Controlling cutting pattern changes in one dimensional trim problems. Operations Research, 23(3), 483-493.

R. W. HAESSLER (1979). Solving the two-stage cutting-stock problem. Omega, The International Journal of Management Science, 7(2), 145-151.

R. W. HAESSLER (1980). A note on computational modifications to the Gilmore-Gomory cutting stock algorithm. Operations Research, 28(4), 1001-1005.

A. I. HINXMAN (1980). The trim-loss and assortment problems: a survey. European Journal of Operational Research, 5, 8-18.

R. HOTO (2001). O problema da mochila compartimentada aplicada no corte de bobinas de aço. Doctor Thesis, COPPE/UFRJ, Engineering of Systems and Computation, Rio de Janeiro, R.J., Brazil.

R. HOTO, N. MACULAN, M. N. ARENALES and F. P. MARQUES (2002). Um novo procedimento para o cálculo de mochilas compartimentadas. Investigação Operacional-APDIO, 22, 213-234.

R. E. JOHNSTON and L. R. KHAN (1995). Bounds for nested knapsack problems. European Journal of Operational Research, 81, 154-165.

E. Yu-Hsien LIN (1998). A bibliographical survey on some well-known non-standard knapsack problems. INFOR, 36(4), 274-317.

S. MARTELLO and P. TOTH (1990). Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, Chichester.

D. PISINGER (1995). Algorithms for knapsack problems. Ph.D. Thesis, Department of Computer Science, University of Copenhagen, Copenhagen, Denmark.

H. SUZUKI (1978). A generalized knapsack problem with variable coeficientes. Mathematical Programming, 15, 162-176.

H. STADTLER (1990). A one dimensional cutting-stock problem in the aluminium industry and its solution. European Journal of Operational Research, 44, 209-223.

N. Y. SOMA, H. H. YANASSE, A. S. I. ZINOBER and P. J. HARLEY (1995). A polynomial approximation scheme for the subset sum problem. Discrete Applied Mathematics, 57, 243-253.

G. WÄSCHER and T. GAU (1996). "Heuristics for the integer one dimensional cutting-stock problem: a computacional study." OR Spektrum, 18, 131-144.

H. H. YANASSE, N. Y. SOMA (1987). A new enumeration scheme for the knapsack problem. Discrete Applied Mathematics, 18, 235-245.

H. H. YANASSE, N. Y. SOMA and N. MACULAN (2000). An algorithm for determining the k-best solutions of one-dimensional knapsack problem. Pesquisa Operacional, 20(1), 117-134.

E. J. ZAK (2002). Row and column generation technique for a multistage cutting stock problem. Computers & Operations Research, 29, 1143-1156.

Table 1. Numerical results of random instances of the CCSP.

Instances		Combination 1 COMP_bound COMPREST_bound			Combination 2 COMP_bound COMPREST			Combination 3 COMP_ex COMPREST_bound			Combination 4 COMP_ex COMPREST		
Classes	items	time	waste	interm	time	waste	interm	time	waste	interm	time	Waste	interm
	32	00:01:993	< 0,1%	773	00:03:725	< 0,1%	761	00:09:283	< 0,1%	587	00:08:402	< 0,1%	577
	32	00:03:445	0,7%	748	00:04:467	0,7%	748	00:07:761	< 0,1%	552	00:09:924	< 0,1%	536
5	27	00:02:794	0,5%	710	00:02:524	0,5%	704	00:05:718	< 0,1%	512	00:07:881	< 0,1%	511
	32	00:02:885	0,6%	792	00:03:685	0,6%	788	00:05:208	< 0,1%	695	00:05:919	< 0,1%	685
	35	00:03:836	1,2%	806	00:03:715	1,2%	791	00:06:099	< 0,1%	589	00:07:301	< 0,1%	578
	62	00:08:773	< 0,1%	1475	00:17:815	< 0,1%	1478	00:14:711	< 0,1%	1270	00:23:514	< 0,1%	1283
	62	00:27:069	< 0,1%	1335	00:28:972	< 0,1%	1296	00:18:827	< 0,1%	1294	00:32:567	< 0,1%	1270
10	52	00:07:631	< 0,1%	1231	00:09:344	< 0,1%	1233	00:09:914	< 0,1%	1197	00:13:259	< 0,1%	1208
	60	00:11:466	0,4%	1483	00:22:903	0,4%	1496	00:22:242	< 0,1%	1179	00:23:574	< 0,1%	1176
	57	00:03:996	1,0%	1347	00:14:912	1,0%	1376	00:17:896	< 0,1%	1113	00:26:619	< 0,1%	1117
	67	00:09:694	< 0,1%	1577	00:14:621	< 0,1%	1586	00:22:363	< 0,1%	1341	00:32:076	< 0,1%	1356
	85	00:20:860	< 0,1%	2010	00:32:066	< 0,1%	2007	00:57:603	< 0,1%	1576	01:15:438	< 0,1%	1575
15	82	00:15:732	0,7%	1948	00:34:881	0,7%	1957	00:33:999	< 0,1%	1545	00:34:680	< 0,1%	1538
	82	00:16:283	< 0,1%	2122	00:18:046	< 0,1%	2126	00:39:146	< 0,1%	1637	00:53:076	< 0,1%	1648
	75	00:12:358	2,8%	1660	00:17:325	2,8%	1639	00:34:710	< 0,1%	1433	00:54:398	< 0,1%	1422
	95	00:24:505	1,0%	2136	00:30:294	1,1%	2146	01:16:070	< 0,1%	1984	01:23:230	< 0,1%	1989
	110	00:44:775	1,3%	2491	01:15:288	1,3%	2501	02:23:306	< 0,1%	1999	02:23:377	< 0,1%	1974
20	120	00:47:087	4,1%	2644	02:07:103	4,1%	2655	02:00:333	< 0,1%	2565	02:27:833	< 0,1%	2572
	102	00:23:964	1,2%	2286	00:51:083	1,2%	2286	01:20:425	< 0,1%	2182	02:04:509	< 0,1%	2205
	100	00:59:856	< 0,1%	2469	01:08:078	< 0,1%	2484	01:04:903	< 0,1%	2029	01:30:790	< 0,1%	2067
30	142	02:14:623	< 0,1%	3598	02:46:780	< 0,1%	3602	04:34:454	< 0,1%	3184	06:47:075	< 0,1%	3218
40	207	05:23:214	0,4%	4946	11:25:415	0,4%	5008	13:27:371	< 0,1%	4456	14:26:066	< 0,1%	4452

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

081/2004 HOTO, R.; ARENALES, M.; MACULAN, N. – The compartmentalized knapsack problem: a case study.

080/2004 KAIBARA, M.K.; FERREIRA, V.G.; NAVARRO, H.A. – Upwinding finite-difference schemes for convection dominated problems – Part I: theoretical results.

079/2004 PAIVA, D. M. B.; FREIRE, A. P.; FORTES, R. P. M. – Web engineering process – a case study from academic development

078/2004 SOUZA, R.; SILVA, C.; ARENALES, M.N. – Método do tipo dual simplex para problemas de otimização linear canalizados: teoria

077/2004 SILVA, A M.P.; NUNES, M.G.V. - Using multiword lists for lexically aligning brazilian portuguese and english texts..

076/2004 BÍSCARO, H.H; CASTELO FILHO, A.; NONATO, L.G. – A topological approach to curve reconstruction from scattered points.

075/2004 NONATO, L.G.; CASTELO FILHO, A.; CAMPOS, J.E.P.P.; BÍSCARO, H.; MINGHIM, R. – Topological tetrahedron characterization with applications in volumetric reconstruction.

074/2003 TOMÉ, M.F.; CUMINATO, J.A.; CASTELO, A.; FERREIRA, V.G.; McKEE, S. - Recent Developments in the MAC Technique.

073/2003 SOUZA, F.S.; MANGIAVACCHI, N.; NONATO, L.G.; CASTELO FILHO, A.; TOMÉ, M.F.; FERREIRA, V.G.; CUMINATO, J.A.; McKEE, S. – A front-tracking finite difference method to solve the 3D navier-stokes equations for multi-fluid flows with free surfaces.

072/2003 MANGIAVACCHI, N.; CASTELO FILHO, A.; TOMÉ, M.F.; CUMINATO, J.A.; OLIVEIRA, M.L.B.; McKEE, S. – A numerical technique for incluiding surface tension effects for axisymmetric and planar flows using the gensmac method.

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

081/2004 HOTO, R.; ARENALES, M.; MACULAN, N. – The compartmentalized knapsack problem: a case study.

080/2004 KAIBARA, M.K.; FERREIRA, V.G.; NAVARRO, H.A. – Upwinding finite-difference schemes for convection dominated problems – Part I: theoretical results.

079/2004 PAIVA, D. M. B.; FREIRE, A. P.; FORTES, R. P. M. – Web engineering process – a case study from academic development

078/2004 SOUZA, R.; SILVA, C.; ARENALES, M.N. – Método do tipo dual simplex para problemas de otimização linear canalizados: teoria

077/2004 SILVA, A M.P.; NUNES, M.G.V. - Using multiword lists for lexically aligning brazilian portuguese and english texts..

076/2004 BÍSCARO, H.H; CASTELO FILHO, A.; NONATO, L.G. – A topological approach to curve reconstruction from scattered points.

075/2004 NONATO, L.G.; CASTELO FILHO, A.; CAMPOS, J.E.P.P.; BÍSCARO, H.; MINGHIM, R. – Topological tetrahedron characterization with applications in volumetric reconstruction.

074/2003 TOMÉ, M.F.; CUMINATO, J.A.; CASTELO, A.; FERREIRA, V.G.; McKEE, S. - Recent Developments in the MAC Technique.

073/2003 SOUZA, F.S.; MANGIAVACCHI, N.; NONATO, L.G.; CASTELO FILHO, A.; TOMÉ, M.F.; FERREIRA, V.G.; CUMINATO, J.A.; McKEE, S. – A front-tracking finite difference method to solve the 3D navier-stokes equations for multi-fluid flows with free surfaces.

072/2003 MANGIAVACCHI, N.; CASTELO FILHO, A.; TOMÉ, M.F.; CUMINATO, J.A.; OLIVEIRA, M.L.B.; McKEE, S. – A numerical technique for incluiding surface tension effects for axisymmetric and planar flows using the gensmac method.