A decomposition problem of a natural number for a rectangular cutting-covering model

PAUL IACOB
Transilvania University of Braşov
Department of Computer Science
Iuliu Maniu 50, 500091 Brasov
ROMANIA

DANIELA MARINESCU
Transilvania University of Braşov
Department of Computer Science
Iuliu Maniu 50, 500091 Brasov
ROMANIA

KINGA KISS-IAKAB
Transilvania University of Braşov
Department of Computer Science
Iuliu Maniu 50, 500091 Brasov
ROMANIA

Abstract: We study in this paper a cutting-covering problem, defined by us in [2] and [3], the problem of covering a rectangular support with rectangular pieces cut from a roll. We first prove that the algorithm used in [2] and [3] for the rectangular cutting-covering problem without losses is not optimal. Starting from a decomposition of a natural number in sums of naturals we develop an algorithm for a better solution for the rectangular cutting-covering problem. Some examples and also the estimation of this algorithm's complexity are presented.

Key–Words: rectangular cutting, covering, optimization

1 Introduction

Cutting and Covering problems belong to an old and very well-known family. Such problems are also commonly referred to as Cutting and Packing problems, called CP in [6]. This is a family of natural combinatorial optimization problems, admitted in numerous real world applications from computer science, industrial engineering, logistics, manufacturing, management, production process, etc. If we know the dimensions of the pieces then we are dealing with a classical cutting-stock problem, which can be modelled as a mixed 0-1 programming problem [4]. There are also heuristic models as [7].

Cutting-stock problems have many applications in production processes of paper, glass, metal and timber cutting industries. Many different versions of cutting problems exist as there are one, two and three dimensional cuttings with or without several constraints. Such a constraint is the guillotine restriction where one can separate the pieces from the cutting model through a number of guillotine cuts (i.e. cuts through the 2 whole sub-models).

The two-dimensional covering problem is related to the two-dimensional cutting-stock problem, which is also richly documented.

2 Problem Formulation

We study in this paper a new cutting-covering problem, defined by us in [1] and [2], where we don't know the dimensions of pieces for covering.

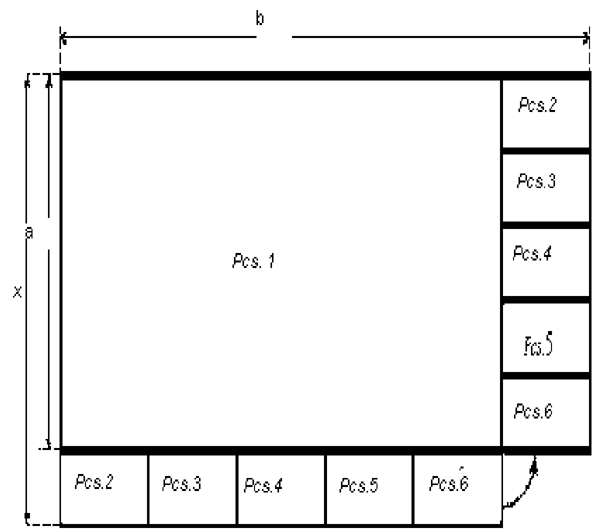

Figure 1: Covering the rectangle with 6 pieces

From the material in a roll of fixed width and infinite length, we cut rectangular pieces by guillotine-cuts, so that we cover a rectangle of dimensions a, b, without losses or overlapping and with a minimum number of pieces.

In [1] and [2] we gave an algorithm due to which a rectangle of dimensions $a = 5$, $b = 50$ can be covered by a roll of the width $x = 6$, with six pieces, as it follows:

We have considered the following problem: Can we cover the support with fewer pieces? One solution is the one from the Figure 2.

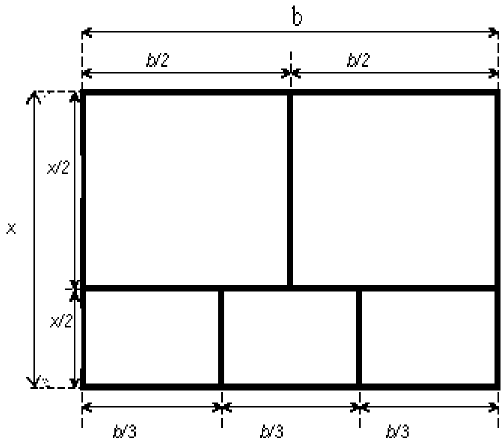

Figure 2: Covering a rectangle of dimensions *a* x *b* with five pieces

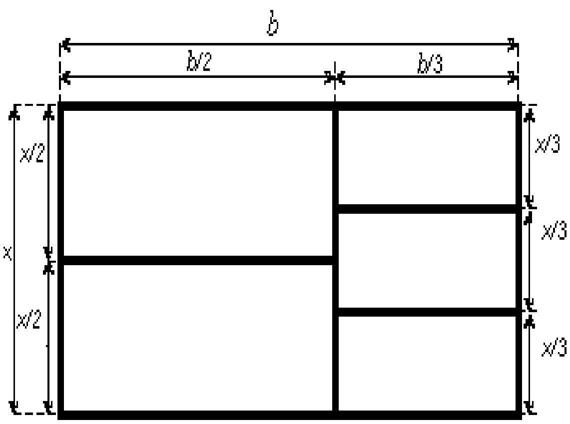

Figure 3: Cutting of the five pieces for the covering of the rectangle of dimensions *a* x *b*

The solution is found by cutting the material from the roll of width *x* = 6 as shown in the Figure 3.

We can see that we can effectuate the covering only with five pieces and the solution does not depend on the size of *b*, but only on the ratio $\frac{a}{x} = \frac{5}{6}$.

3 Decompositions of a natural number in sums for cutting-covering models

For the cutting of the six pieces, we decompose the natural number 6 in the following eight sums:

$$1+1+1+1+1+1 \quad (1)$$

$$1+1+1+1+2 \quad (2)$$

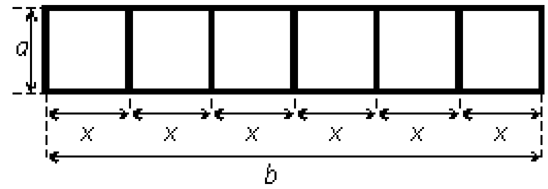

Figure 4: Case (1) *b* = 6*x*

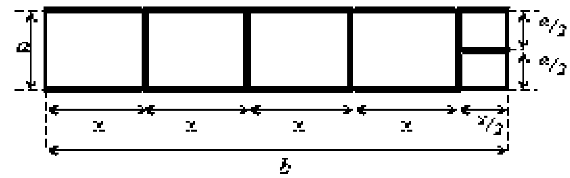

Figure 5: Case (2) $b = \frac{9x}{2}$

$$1+1+1+3 \quad (3)$$

$$1+1+4 \quad (4)$$

$$1+2+3 \quad (5)$$

$$1+5 \quad (6)$$

$$3+3 \quad (7)$$

$$6 \quad (8)$$

Every decomposition gives us a cutting-covering model with six pieces as in the following:

Now we cut pieces of the width *x* first and what really counts is the fractional part.

Decomposition has the form:

$$\left(\underbrace{1, 1, ..., 1}_{s \text{ times}}, \underbrace{p_1, p_2, ..., p_n}_{\text{residual part}} \right) \quad (9)$$

We presume that

$$b = x \left(s + \sum_{i=1}^{n} p_i \right)$$

then the cutting-covering scheme will be the one from Figure 12.

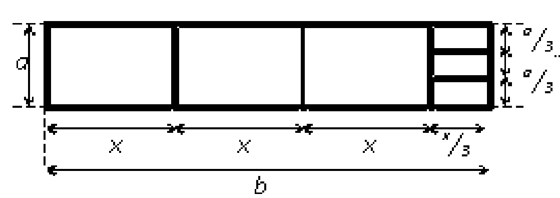

Figure 6: Case (3) $b = \frac{10x}{3}$

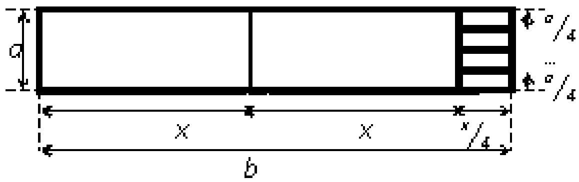

Figure 7: Case (4) $b = \frac{9x}{4}$

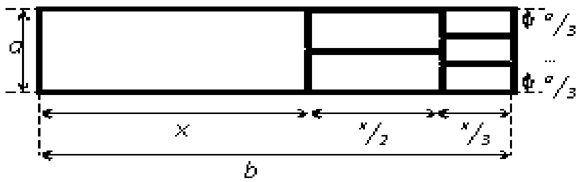

Figure 8: Case (5) $b = \frac{11x}{6}$

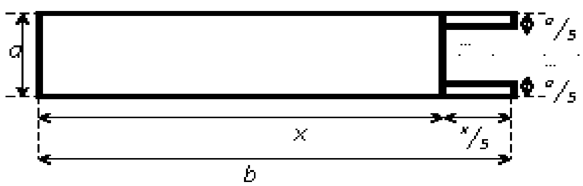

Figure 9: Case (6) $b = \frac{6x}{5}$

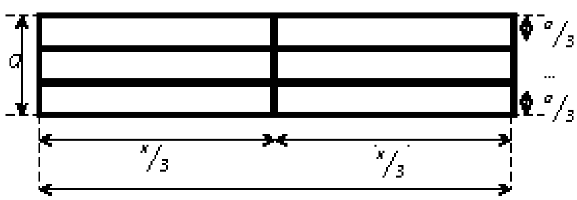

Figure 10: Case (7) $b = \frac{2x}{3}$

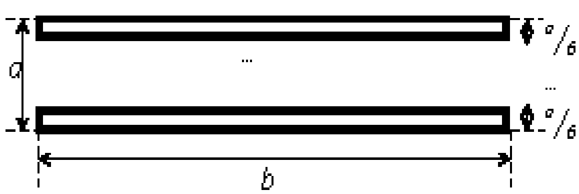

Figure 11: Case (8) $b = \frac{x}{6}$

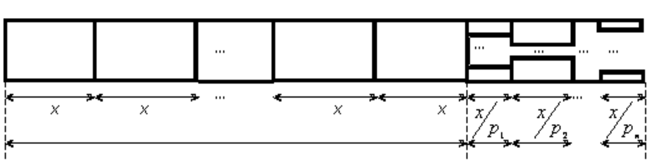

Figure 12: The general covering scheme

We note with A the pieces of length x and with B_i the pieces of length p_i, for i = 1, 2, ..., n. The covering model from figure 12 is a model with guillotine restrictions and due to [5], can be represented as a word of a pictural language, as it follows:

$$\underbrace{A \oplus A \oplus ... \oplus A}_{s \text{ times}} \oplus \underbrace{(B_1 \ominus B_1 \ominus ... \ominus B_1)}_{p_1 \text{ times}} \oplus \underbrace{(B_2 \ominus B_2 \ominus ... \ominus B_2)}_{p_2 \text{ times}} \oplus ... \oplus \underbrace{(B_n \ominus B_n \ominus ... \ominus B_n)}_{p_n \text{ times}}$$

In the formula, we note $\oplus$ the column concatenation and $\ominus$ the row concatenation, between two rectangles.

4 Algorithms for finding the model with minimum number of pieces and without losses

We observe that an optimal model depends only on the fact that $\frac{a}{x}$ or $\frac{b}{x}$ can be written as

$$\left(s + \sum_{i=1}^{n} p_i\right)$$

where p_i are ascending sorted integers, p_i appears at most p_i - 1 times, and the covering will contain

$$\left(s + \sum_{i=1}^{n} p_i\right)$$

pieces. Because s is not important for the matching, it results that it is required to have one of the following equalities:

$$\mathscr{F}r\left(\frac{a}{x}\right) = \mathscr{F}r\left(\sum_{i=1}^{n} \frac{1}{p_i}\right) \quad (10)$$

$$\mathscr{F}r\left(\frac{b}{x}\right) = \mathscr{F}r\left(\sum_{i=1}^{m} \frac{1}{p_i}\right) \quad (11)$$

where we note $\mathscr{F}r\left(\frac{a}{x}\right)$ the fractional part of $\frac{a}{x}$. We will use the following notations:

- the row vectors $S[i] = (p_1^i, ..., p_{n_i}^i)$ for fractional parts of a decomposition like (9)
- the column vector

$$R[i] = \sum_{j=1}^{n_i} \frac{1}{p_j^i}$$

- the column vector

$$N_b[i] = \sum_{j=1}^{n_i} \frac{1}{p_j^i}$$

  for the number of fractional pieces

- $\mathscr{S}$ the matrix of vectors *S*[*i*], where each *S*[*i*] is completed by zero components until the dimensions max$\{n_i\}$.

We present first an algorithm for creating a database that will be used in the algorithm for finding the model with a minimum number of pieces and without losses.

4.1 Algorithm *Datab*

The ***Datab*** algorithm will create one table for each *N*, the number of the fractional pieces.

Input data: N_{max}, the maximum number of the fractional pieces.

Output data: the database.

for *N* = 2 to N_{max} do
{

- generate $(p_1^i, ..., p_{n_i}^i)$, for *i* = 1, 2, . . ., the sequence of fractional parts *S*[*i*] = so that the residual part of the decomposition in sums to have at most *N* pieces.
- generate the two vectors *R* and N_b and the matrix $\mathscr{S}$
- generate the $\mathscr{T}$ matrix by adding the column vectors *R* and N_b to matrix $\mathscr{S}$.
- sort the rows of the matrix $\mathscr{T}$, on asccending mode by 2 keys:
  - the first key - the components of the vector *R*,
  - the second key - the components of the vector N_b.
- generate the reduced matrix $\mathscr{T}'$ by the following operations:
  - elimination from $\mathscr{T}$ of the rows with *R*[*i*] = 0,
  - elimination from $\mathscr{T}$ of the repetitions in *R*, storing only the row corresponding to the component with minimum $N_b[i]$ for the same *R*[*i*].
- memorize in a database of the matrix $\mathscr{T}'$ as $\mathscr{T}_N'$, of the dimensions r_N and c_N.

}

Using this database created by the algorithm ***Datab*** we can define the algorithm for finding the rectangular cutting-covering model with the minimum number of pieces and without losses.

4.2 Algorithm *MinMod*

Input data:
-the dimensions *a* and *b* of the supporting plate and the dimension *x*, the width of the material roll;
-the databases created by ***Datab*** algorithm.

Output data:
-*N*;
-the matrix $\mathscr{T}_{\mathcal{N}'}$, representing the cutting-covering model.

N = 2;
repeat
{
  k = 0;
  for *i* = 1 to r_N do
  {
    if (*R*(*i*) = $\mathscr{F}r\left(\frac{a}{x}\right)$ or *R*(*i*) = $\mathscr{F}r\left(\frac{b}{x}\right)$)
      then *k* = 1;
  }
  N = *N* + 1;
}
until(*k* = 1 or $N > N_{max}$);
if $N > N_{max}$ then
  write ”there is no cutting-covering
    model with at most N_{max} pieces”
  else
    write the model
    corresponding to (1)-(8)

4.3 Examples

Let us consider 2 practical examples of cutting-covering problems.

Example 1. Let the rectangular support of dimensions *a* = 3 and *b* = 5 and the material from a roll of the width *x* = 2. We first present an example of the ***Datab*** algorithm for N=8. In this case the sequences S_i will be
S_1 = (2),
S_2 = (3),
S_3 = (4),
S_4 = (5),

S_5 = (6),
S_6 = (7),
S_7 = (8),
S_8 = (2, 3),
S_9 = (2, 4),
S_{10} = (2, 5),
S_{11} = (2, 6),
S_{12} = (3, 3),
S_{13} = (3, 4),
S_{14} = (3, 5),
S_{15} = (4, 4),
S_{16} = (2, 3, 3),
and then we have the two vectors
N_b = (2, 3, 4, 5, 6, 7, 8, 5, 6, 7, 10, 6, 7, 8)
and
$R = \left(\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \frac{5}{6}, \frac{3}{4}, \frac{7}{10}, \frac{2}{3}, \frac{7}{12}, \frac{8}{15}, \frac{2}{4}, \frac{1}{6}\right).$
After sorting by *R* and eliminations like R_{12}, R_{15}, R_{16} we obtain the Table 1:

N_b	*R*		*S*	
8	$\frac{1}{8}$	8	0	0
7	$\frac{1}{7}$	7	0	0
6	$\frac{1}{6}$	6	0	0
5	$\frac{1}{5}$	5	0	0
4	$\frac{1}{4}$	4	0	0
3	$\frac{1}{3}$	3	0	0
2	$\frac{1}{2}$	2	0	0
8	$\frac{8}{15}$	3	5	0
7	$\frac{7}{12}$	3	4	0
6	$\frac{2}{3}$	3	3	0
7	$\frac{7}{10}$	2	5	0
6	$\frac{3}{4}$	2	4	0
5	$\frac{5}{6}$	2	3	0

Table 1

Now, with this database, we can apply the algorithm ***MinMod*** for our cutting-covering problem.

If *a* = 3, *b* = 5 and *x* = 2
then
$\frac{a}{2} = 1 + \frac{1}{2}$ and $\frac{b}{2} = 2 + \frac{1}{2}$.

Now we have the same fractionary part, but $\frac{b}{2}$ gives more pieces. So the model will have 1+2 = 3 pieces two of these being fractionar. In the database table for *R* = $\frac{1}{2}$ the decomposition is *S* = (2), which means we have two pieces of the width $\frac{x}{2}$. The covering model is the model from the Figure 13.

Example 2. If *a* = 5.4, *b* = 2.3 and *x* = 2 to obtain an integer we change the measurement unit to one 10 times smaller than the previous one and we have
a' = 54, *b'* = 23 and *x* = 20

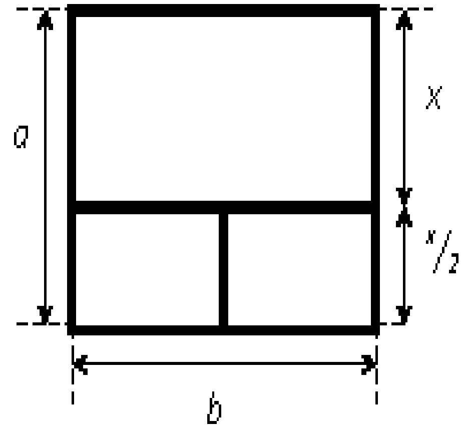

Figure 13: Covering scheme

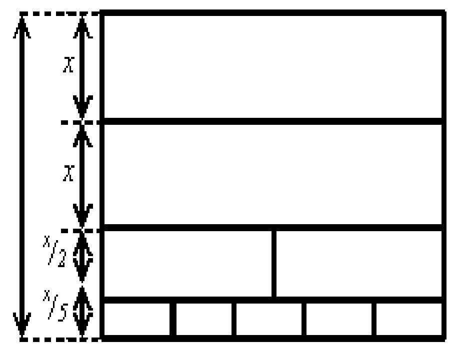

Figure 14: Covering scheme

then
$\frac{a'}{20} = 2 + \frac{7}{20}$ and $\frac{b'}{20} = 1 + \frac{3}{20}$.

So the model will have 2 pieces of the width *x* and 7 fractional pieces, totally 2+7 = 9 pieces.

In the database table for $\frac{R}{7}$ the decomposition is *S* = (2,5), wich means we have 2 pieces of the width $\frac{x}{2}$ and 5 fractional pieces of the width $\frac{x}{5}$. The covering model is the model from the Figure 14.

We observe that the models obtained in this way are symmetric. Symmetrization techniques can be found in [1].

5 Complexity

The rows of the matrix *S* are 1, 2, 3, ... , *N* - combinations from the set with *N* - 1 elements. Let C'^k_N the combination with repetition, where *N* is the number of objects out of which you can choose and *k* is the number to be chosen. So the construction of this matrix will have the complexity:

$$T(N) \leq C'^1_{N-1} + C'^2_{N-1} + C'^3_{N-1} + ... + C'^{N-1}_{N-1}$$

$$= C_N^1 + C_{N+1}^2 + C_{N+2}^3 + ... + C_{2N-2}^N$$
$$< C_{2N-2}^1 + C_{2N-2}^2 + C_{2N-2}^3 + ... + C_{2N-2}^N$$
$$= \frac{1}{2}\left(2^{2N-2} - 2 + C_{2N-2}^N\right)$$
$$< 2^{2N-3}$$

Obviously this complexity is non-polynomial, but the construction is made only once and then we memorize it in the database. The Divide et Impera searching in the *R* vector will have the complexity of $log_2(2^{2N-3}) = 2N-3$, so liniar depending on *N* - the number of maximum fractional pieces.

6 Conclusions

The cutting-covering model introduced by us in [1, 2] is a different one to the classical models because at the beginning, we know nothing about the dimensions of the pieces from the model. We will find these dimensions and, at the same time, the arrangement of the pieces on the support, only after the application of the MinMod algorithm.

The algorithm MinMod is minimal and without losses because if a/x or b/x is not integer then it is necessary to use some fractional pieces. Every fractional piece of the dimension x/p_i is used p_i times and then the used area from the material is a rectangle of the same area like the rectangular support.

References:

[1] Iacob, P., Marinescu, D., Luca, C.: L-Shape room. Proceeding of WMSCI, Orlando, Florida, USA, 2005, Vol III, pp. 175-179.

[2] Iacob, P., Marinescu, D., Luca, C.: Covering a rectangle with rectangular pieces. Proccedings - ”Logistics from α to Ω: Strategies and Applications”, Thessaloniki, Greece, October 18-20, 2001, pp 506-513.

[3] Iacob, P., Marinescu, D.: Upon a cutting problem with two optimization criteria. Conference on analysis, functional, equations, approximation and convexity Cluj-Napoca, October 15-16, 1999;pp.84-88.

[4] Kalpic, D., Mornar, V., Fertal, K.: An algorithm for corrugated paper cutting, Proceedings of WSEAS Conference, 2002, pp. 1651-1655.

[5] Marinescu, D.: A s-picture language for a cutting-stock model with guillotine restrictions. Buletin of the Transilvania University of Brasov - seria C, Vol XXXIII 1991, pp 39-45.

[6] Sweeney, P., Paternoster, E.: Cutting and packing problems: a categorized application-oriented research bibliography. Journal of the Operational Research Society, 43/7 (1992) 691-706.

[7] Xiang Song, Cheng-Bin Chu, Yi-Yong Nie: A Heuristic Dynamic Programming Algorithm for 2D Unconstrained Guillotine Cutting, Proceedings of WSEAS Conference, 2004, paper 484-108.