

A NEW MATHEMATICAL MODEL FOR THE CUTTING STOCK/LEFTOVER PROBLEM

Marcos Arenales¹, Adriana Cherri^{2*}, Douglas N. do Nascimento³ and Andréa Vianna⁴

*Av. Eng.º. Luiz Edmundo Carrijo Coube, 14-01, Bauru, SP, Brasil
Postal address: 17033-360, phone +55-14-3103-6086

¹Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo

²Departamento de Matemática, Universidade Estadual Paulista “Júlio de Mesquita Filho”

^{3,4}Departamento de Computação, Universidade Estadual Paulista “Júlio de Mesquita Filho”

E-mails: arenas@icmc.usp.br¹, adriana@fc.unesp.br²,
douglasnn@fc.unesp.br³, vianna@fc.unesp.br⁴

ABSTRACT

This paper addresses the cutting stock/leftover problem (CSLP), which differs from the ordinary cutting stock problem (CSP) by retaining stock leftovers that can be cut in the future to meet new demands. Therefore, leftovers are not considered waste in the current period. A new mathematical model for the CSLP is presented to capture a well-used strategy in the practice of cutting, which consists of partially cutting the objects in stock, and keeping the leftovers to be cut in the next periods. Computational experiments were made for the one-dimensional case, although other dimensions can be considered straightforward.

Keywords: cutting stock problem, usable leftovers, mathematical model, column generation.

1 Introduction

Cutting Stock Problems (CSP) are central stages in production planning for a number of industries that have to cut paper rolls, steel bobbins, steel bars, wooden hardboards, leather pieces, among others. Basically, CSP consists of cutting large pieces (*objects*) available in stock into a set of smaller pieces (*items*) with specified quantities and sizes by optimizing an objective function, such as minimizing the total length/area/volume of the objects cut, minimizing the total waste or minimizing the cost of the objects cut.

Although there are many articles in the literature on general CSP, many constraints arise in practical situations that lead to the need of developing new mathematical models and new approaches to solve them. Among the various types of CSP in the literature (see Wäscher *et al.*(2007), for a typology), a problem which has not been fully studied yet, although it often appears in practice, consists of considering usable leftovers when the objects are cut. These leftovers are kept in stock to meet future demands, and therefore they are not wasted. We call this problem the Cutting Stock/Leftover Problem (CSLP) and we may refer to it as 1D-CSLP if only one dimension is relevant in the cutting process, such as cutting bars and rolls.

A considerable amount of literature has been published on 1D-CSLP, see Cherri *et al.* (2014) for a survey. To the best of our knowledge, it was first mentioned by Brown (1971), but only to differentiate preferable solutions among others with some waste. Scheithauer (1991) proposed the first mathematical model for CSLP, which is a cutting pattern-oriented model (see Kantorovich (1960) and Gilmore and Gomory (1961)). The main idea in the Scheithauer model consists of considering fictitious items (or residual lengths with no demand associated) that can be cut in the future. These fictitious items play the role of leftovers with special utility values. This approach reduces the waste by diversifying items (it is well-known that instances with a diversity of sizes of items have less waste). We call this as an *item-oriented approach*. Cui and Yang (2010) extended the Scheithauer model to include upper bounds on the number of leftovers.

An *assignment approach* is given by Abuabara and Morabito (2009), in which assigning or not an item to an object is a decision to be made. This approach is able to find optimal solutions for smaller instances. However, most of the methods in the literature are greedy heuristics that perturb a good solution of the CSP in order to try to join waste together (Roodman (1986), Gradisar *et al.* (1997), Gradisar *et al.* (1999), Cherri *et al.* (2009), Cherri *et al.* (2013), among others). In addition, they define bounds to determine the minimum acceptable length for the generated leftovers during the cutting process (see Cherri *et al.* (2014) for a survey). Thus, a great diversity of lengths of leftovers can be generated, which may be unacceptable to many industries since this needs efficient control systems to retrieve them in the future, and a high risk of being wasted as well. This is not a conjecture, the authors have already witnessed this situation. In our approach to this paper, the same leftover types may always be found in stock, as the manager decides the sizes and quantities of the leftovers.

Andrade *et al.* (2013) and Andrade *et al.* (2014) presented multilevel MIP models to solve the two-dimensional cutting stock problem with usable leftovers. They used bounds to define leftover dimensions not allowing a control on the number of leftover types.

Cherri *et al.* (2014) reviewed papers from the literature that consider the 1D-CSLP. In this survey, the authors present applications of the problems, the mathematical model (if it was proposed), comments on the results obtained from each paper and proposals for future studies related to the 1D-CSPUL.

In this paper, an object-oriented mathematical model is proposed to solve CSLP. This new approach takes advantage of diversifying objects as it is well-known that more sizes in stock lessen the waste. To model this, *reduced objects* which include one leftover are given to the pricing problem in the cutting pattern-oriented model. This is a promising strategy to solve CSLP as the lengths of the possible leftovers are previously defined and the advantages of generating or not leftovers can be analysed using the model, *i.e.*, the model search for cutting patterns that include or not leftovers, so that it works as if a number of different object sizes were available.

The remainder of the paper is outlined as follows. In Section 2, the CSLP is defined and a mathematical model to solve it is proposed. The concepts concerning our strategy to solve the problem are also well quantified in this section. Section 3 describes the solution methods and computational results are given in Section 4. In Section 5, we present some concluding remarks.

2 The CSLP Object-Oriented Model

In the CSLP, a set of items must be produced by cutting large objects of standard sizes (objects bought on the market) or nonstandard (objects that are leftovers of previous cuts). Similarly to the CSP, there are given demands and sizes of items on one hand, and availabilities and sizes of objects, on the other. Demands must be met by cutting the available objects in stock in order to minimize waste, or another suitable objective function. However, new leftovers of given sizes can be generated within a limited amount, which are not counted as waste.

In summary, we consider the following practical assumptions:

1. Any *standard object* can be completely cut or partially cut to generate two new objects: a *reduced object* to be currently cut and a *leftover* to be kept in stock for future demands;
2. Any leftover has its size given beforehand;
3. The total number of new leftovers is limited;

Data:

- S : number of types of standard objects. We denote object type s , $s \in \{1, \dots, S\}$;
 R : number of types of leftovers in stock. We denote leftover type $S+s$, $s \in \{1, \dots, R\}$
 e_s : number of objects/leftover type s available in stock, $s = 1, \dots, S + R$;
 L_s : length of object/leftover type s , $s = 1, \dots, S + R$;
 m : number of types of ordered items;
 d_i : demand for item type i , $i = 1, \dots, m$;
 ℓ_i : length of item type i , $i = 1, \dots, m$;
 J_s : set of cutting patterns for object type s , $s = 1, \dots, S + R$;
 $J_s(k)$: set of cutting patterns for standard object type s with leftover type $S+k$, $k = 1, \dots, R$, $s = 1, \dots, S$;
 a_{ijs} : number of items type i in cutting pattern j for object type s , $i = 1, \dots, m$, $s = 1, \dots, S + R$, $j \in J_s$;
 a_{ijsk} : number of items type i in cutting pattern j for object type s and leftover type $S+k$, $i = 1, \dots, m$, $k = 1, \dots, R$, $s = 1, \dots, S$, $j \in J_s(k)$;
 c_{js} : waste of cutting object/leftover s according to pattern j , $s = 1, \dots, S + R$, $j \in J_s$;
 c_{jsk} : waste of cutting object s according to pattern j when generating a leftover type $S+k$, $s = 1, \dots, S$, $k = 1, \dots, R$, $j \in J_s(k)$;
 U : maximum number of leftovers;

Variables:

- x_{js} : number of objects type s cut according to pattern j , $s = 1, \dots, S + R$, $j \in J_s$;
 x_{jsk} : number of objects type s cut according to pattern j generating a leftover type $S+k$, $s = 1, \dots, S$, $k = 1, \dots, R$, $j \in J_s(k)$.

Remarks:

1. The size of a reduced object obtained by partially cutting an object type s with leftover type $S+k$ is given by: $L_s - L_{S+k}$. Only standard objects can be cut with a leftover.
2. Although in Assumption 1 any standard object can be used to generate a leftover, it is possible to allow leftovers of only one type of object.
3. Assumptions 2 and 3 are considered to make operational handling feasible. Leftovers may be limited to only one size.

The CSLP Object-Oriented Model

$$\text{Minimize } f(x) = \sum_{s=1}^S \sum_{j \in J_s} c_{js} x_{js} + \alpha' \sum_{s=1}^S \sum_{k=1}^R \sum_{j \in J_s(k)} c_{jsk} x_{jsk} + \alpha'' \sum_{s=S+1}^{S+R} \sum_{j \in J_s} c_{js} x_{js} \quad (1)$$

subject to:

$$\sum_{s=1}^S \sum_{j \in J_s} a_{ijs} x_{js} + \sum_{s=1}^S \sum_{k=1}^R \sum_{j \in J_s(k)} a_{ijsk} x_{jsk} + \sum_{s=S+1}^{S+R} \sum_{j \in J_s} a_{ijs} x_{js} = d_i, \quad i = 1, \dots, m \quad (2)$$

$$\sum_{j \in J_s} x_{js} + \sum_{k=1}^R \sum_{j \in J_s(k)} x_{jsk} \leq e_s, \quad s = 1, \dots, S \quad (3)$$

$$\sum_{j \in J_s} x_{js} \leq e_s, \quad s = S+1, \dots, S+R \quad (4)$$

$$\sum_{s=1}^S \sum_{k=1}^R \sum_{j \in J_s(k)} x_{jsk} - \sum_{s=S+1}^{S+R} \sum_{j \in J_s} x_{js} \leq U - \sum_{s=S+1}^{S+R} e_s \quad (5)$$

$$x_{js} \geq 0, s = 1, \dots, S+R, j \in J_s; x_{jsk} \geq 0, k = 1, \dots, R, s = 1, \dots, S, j \in J_s(k) \text{ and integer (6)}$$

Factors $\alpha' \geq 1$ and $\alpha'' \leq 1$ in objective function (1) can be used to say that a new leftover should be created only if it is really attractive, or an older leftover should be cut although it produces more waste than other objects cut.

The objective function (1) to be minimized determines the total waste of cutting standard, with or without leftovers, and nonstandard (previous leftovers) objects. Constraints (2), (3) and (4) are demand and stock constraints, respectively. Constraint (5) limits the net quantity of leftovers, that is, no more than U units of leftovers are allowed for the next period. Non-negativity and integrality of variables x_{js} and x_{jsk} are ensured by the constraint (6).

Because of the exponential number of variables and the integrality conditions, it is hard to solve model (1) – (6) at optimality. The linear relaxation, obtained by relaxing the integer condition in (6), can be easily solved with the column generation technique (Gilmore and Gomory (1963)) and good integer solutions can be found using heuristic procedures (see Wäscher and Gau (1996), Poldi and Arenales (2009), among others). The linear relaxation enables us to analyze tendencies of gains, i.e., how much waste is expected to be diminished by using different policies on leftover stock and how these gains depend on data.

3 Solving the linear relaxation of the CSLP object-oriented model

Before describing the column generation technique (Gilmore and Gomory, 1963, Desaulniers *et al.* 2005) for the CSLP object-oriented model, we show how to price each column. Suppose model (1) – (6) has been solved using a few columns, called the *Restricted Master Problem*, and the simplex multiplier (dual variable) is given by $(\pi^1 \ \pi^2 \ \pi^3 \ \pi^4)$, where π^1 is a m -vector, π^2 a S -vector, π^3 a R -vector, and π^4 a real number, associated to constraints (2) – (5) respectively.

We distinguish three cases:

a) Standard objects, $s \in J_s, s=1, \dots, S$

A general column associated to J_s is given by (index j is omitted):

$$\mathbf{a}_s = (\underbrace{a_{1s} \cdots a_{ms}}_{\substack{m\text{-vector} \\ \text{pattern}}} \mid \underbrace{0 \cdots 1 \cdots 0}_{\substack{S\text{-vector} \\ \text{identity column}}} \mid \underbrace{0 \cdots 0}_{\substack{R\text{-vector} \\ \text{of zeros}}} \mid 0)^T$$

where the first m -vector corresponds to a cutting pattern for a standard object size L_s . Since ℓ_i is the length of item i , $i = 1, \dots, m$, the reduced cost associated to this column is:

$$\begin{aligned} \hat{c}(\mathbf{a}_s) &= L_s - \underbrace{\sum_{i=1}^m \ell_i a_{is}}_{c_s; \text{waste}} - \sum_{i=1}^m \pi_i^1 a_{is} - \pi_s^2 \\ &= - \sum_{i=1}^m (\underbrace{\ell_i + \pi_i^1}_{v_i}) a_{is} + (L_s - \pi_s^2) \end{aligned}$$

b) Reduced objects (or standard objects with leftover), $s \in J_s(k)$, $k = 1, \dots, R$, $s = 1, \dots, S$

Suppose a cut generates a leftover, then a general column associated to $J_s(k)$ is given by (index j is omitted):

$$\mathbf{a}_s(k) = (\underbrace{a_{1s} \cdots a_{ms}}_{\substack{m\text{-vector} \\ \text{pattern}}} \mid \underbrace{0 \cdots 1 \cdots 0}_{\substack{S\text{-vector} \\ \text{identity column}}} \mid \underbrace{0 \cdots 0}_{\substack{R\text{-vector} \\ \text{of zeros}}} \mid 1)^T$$

where the first m -vector corresponds to a cutting pattern for the reduced object $L_s - L_{S+k}$. Therefore, the reduced cost associated to this column is:

$$\begin{aligned} \hat{c}(\mathbf{a}_s(k)) &= \alpha' \left(\underbrace{(L_s - L_{S+k}) - \sum_{i=1}^m \ell_i a_{is}}_{c_s; \text{weighted waste}} - \sum_{i=1}^m \pi_i^1 a_{is} - \pi_s^2 - \pi^4 \right) \\ &= - \sum_{i=1}^m (\underbrace{\alpha' \ell_i + \pi_i^1}_{v_i}) a_{is} + (\alpha' (L_s - L_{S+k}) - \pi_s^2 - \pi^4) \end{aligned}$$

c) Leftover objects, $s \in J_s$, $s = S + 1, \dots, S + R$

A general column associated to J_s , $s = S + 1, \dots, S + R$ is given by (index j is omitted):

$$\mathbf{a}_s = \underbrace{(a_{1s} \cdots a_{ms})}_{\substack{m\text{-vector} \\ \text{pattern}}} \mid \underbrace{(0 \cdots 0)}_{\substack{S\text{-vector} \\ \text{of zeros}}} \mid \underbrace{(0 \cdots 1 \cdots 0)}_{\substack{R\text{-vector} \\ \text{identity column}}} \mid -1)^T$$

where the first m -vector corresponds to a cutting pattern for a leftover L_s . Therefore, the reduced cost associated to this column is:

$$\begin{aligned} \hat{c}(\mathbf{a}_s) &= \underbrace{\alpha''(L_s - \sum_{i=1}^m \ell_i a_{is})}_{c_s: \text{weighted waste}} - \sum_{i=1}^m \pi_i^1 a_{is} - \pi_s^3 + \pi^4 \\ &= - \sum_{i=1}^m \underbrace{(\alpha'' \ell_i + \pi_i^1)}_{v_i} a_{is} + (\alpha'' L_s - \pi_s^3 + \pi^4) \end{aligned}$$

In summary, we have three cases:

a) For a standard object type s : $\hat{c}(\mathbf{a}_s) = - \sum_{i=1}^m v_i a_{is} + (L_s - \pi_s^2)$, where $v_i = \ell_i + \pi_i^1$, and $(a_{1s} \dots a_{ms})^T$ corresponds to a cutting pattern for length L_s ;

b) For a reduced object obtained from a standard object s :

$$\hat{c}(\mathbf{a}_s(k)) = - \sum_{i=1}^m v_i a_{is} + ((\alpha' L_s - L_{S+k}) - \pi_s^2 - \pi^4), \text{ where } v_i = \alpha' \ell_i w_i + \pi_i^1 \text{ and } (a_{1s} \dots a_{ms})^T$$

corresponds to a cutting pattern for $L_s - L_{S+k}$;

c) For a leftover in stock type s :

$$\hat{c}(\mathbf{a}_s) = - \sum_{i=1}^m v_i a_{is} + (\alpha'' L_s - \pi_s^3 + \pi^4), \text{ where } v_i = \alpha'' \ell_i + \pi_i^1, \text{ and } (a_{1s} \dots a_{ms})^T \text{ corresponds}$$

to a cutting pattern to L_s .

The column generation technique is summarized as follows:

Initial step: Build a restricted master problem, *i.e.*, model (1)–(6) with few columns that use trivial cutting patterns, such as homogeneous cutting patterns for standard objects and leftovers. If necessary, include artificial variables. Iteration: $t = 0$.

Step 1. Solve the linear relaxation of the restricted master problem in iteration t .

Step 2.

2.1- For $s = 1, \dots, S$

Solve the cutting problem (v_i is the utility value given in case a))

Maximize $v_1 a_{1s} + \dots + v_m a_{ms}$

subject to: $(a_{1s} \dots a_{ms})$ is a cutting pattern to L_s

calculate the correspondent column and the reduced cost according to case a);

2.2 - For $s = 1, \dots, S$

Solve the cutting problem (v_i is given in case b))

Maximize $v_1 a_1 + \dots + v_m a_m$

subject to: $(a_{1s} \dots a_{ms})$ is a cutting pattern to $L_s - L_{S+k}$, $k = 1, \dots, R$

calculate the correspondent column and the reduced cost according to case b),

2.3 - For $s = S + 1, \dots, S + R$

Solve the cutting problem (v_i is given in case c))

Maximize $v_1 a_1 + \dots + v_m a_m$

subject to that $(a_{1s} \dots a_{ms})$ is a cutting pattern to L_s

calculate the correspondent column and the reduced cost according to case c).

Step 3. Include the most attractive column (*i.e.*, column with *the most negative reduced cost* found in steps 2.1, 2.2 or 2.3) to the restricted master problem. If no attractive column is found, stop – the current solution obtained in Step 1 is an optimal solution to the linear relaxation of model (1)–(6); otherwise, do $t = t + 1$ and go to Step 1.

4 Computational Experiments

To evaluate the performance of the proposed approach in order to analyze waste reduction, 6 classes of instances were considered, by varying lengths and demand of items. For each class, 50 instances were randomly generated. The tests were divided considering two situations. Firstly, we simulated the initial period of the cutting process where there were no leftovers in stock, but considering or not the possibility of generating leftovers. Note that by avoiding the generation of new leftovers together with

no leftover in stock, the problem coincides with the classical cutting stock problem. The second situation simulated an intermediate period with leftovers already available in stock.

For both situations, initial or intermediate periods, the stock only had one type of standard object with length $L_1 = 1000$ and availability large enough to meet the demands. Three types of leftovers were considered with lengths 400, 500 and 600. The availabilities of leftover type $s=1, 2, \text{ and } 3$ were $e_s=0$, for the initial period, and $e_s=k, k=0 \text{ to } 3$, for the intermediate period. The maximum quantity of leftovers varied as $U=3k+3, k=0 \text{ to } 3$. We also considered $\alpha' = \alpha'' = 1$ in the objective function (1), that is, the overall waste to be minimized, not intending to encourage the generation of new leftovers nor to be obliged to use the leftovers in stock.

Each instance had $m=15$ types of demanded items. Length ℓ_i of item type i was randomly generated in the interval $[v_1L_1, v_2L_1]$. We call it *medium item (M)* if $v_1=0.14$ and $v_2=0.4$ (a pattern can contain 3 to 7 items), and *big item (B)* if $v_1=0.3$ and $v_2=0.7$ (a pattern can contain 2 to 3 items). *Small demands (S)* are generated in the interval $[1, 10]$, *medium demands (M)* in the interval $[10, 50]$ and *big demand (B)* in $[50, 300]$. Therefore, a class $[B, S]$ means a set of 50 instances where items are big, and demands are small. The exact values of all the instances we have used can be obtained by contacting the corresponding author.

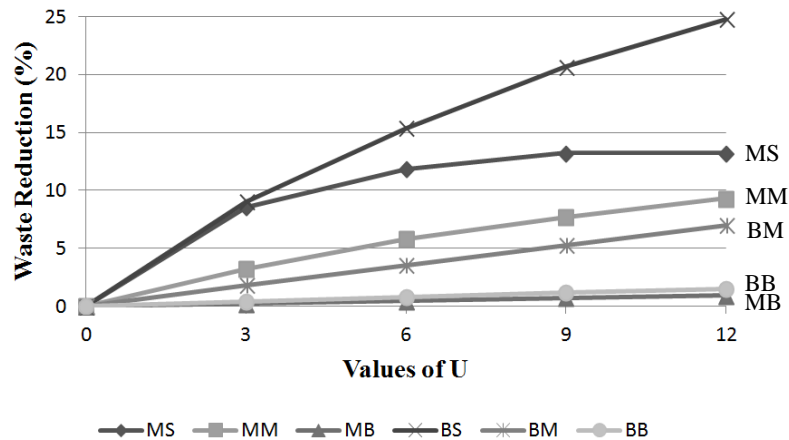
The proposed method was coded using OPL (Optimization Programming Language) of the software CPLEX, version 12.5. The computational tests were run on a computer Core i7, 3.07GHz x 8, 8 GB RAM. The average computation time for each instance was approximately 56 seconds.

Table 1 shows the results for the first period where it is supposed that no leftover is in stock. Note that column $U=0$ consists of the classical cutting stock problem, with no leftovers in stock and no leftover is allowed to be generated. The average waste is given in terms of different values of U . Note that we adopted the total waste to define the objective function, but the total length used could be considered instead. Both are well-used in practice. Graph 1 shows the percentage waste evolution in terms of U related to the classical CSP in the first column in Table 1.

Table 1: Waste for the first period: no leftovers in stock

class	$U = 0$	$U = 3$	$U = 6$	$U = 9$	$U = 12$
[M,S]	136.9	125.1	120.7	118.8	118.8
[M,M]	634.1	613.9	597.5	585.6	575.2
[M,B]	3599.2	3590.3	3581.6	3573.1	3564.7
[B,S]	6971.7	6343.0	5899.7	5530.9	5243.5
[B,M]	39541.7	38830.4	38137.0	37455.0	36787.0
[B,B]	210967.7	210153.7	209353.0	208559.0	207775.0

Graph 1: Percentage of waste for the first period related to $U=0$



These results point out some features. It does not matter if the leftover strategy is used or not when demands are big. This was already expected because the bigger the demands, the more possibilities there are of arranging items in the object so that there is nothing or little to be reduced by generating new leftovers. Of course, the bigger the item sizes, the higher the waste.

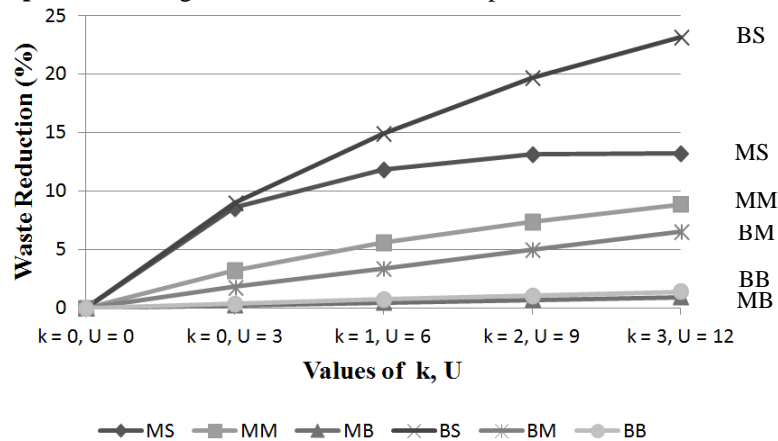
Interesting cases happen when the demands are small – classes [M,S] and [B,S], typical cases of small industries, where the waste is 136.9 and 6971.7, respectively, for the strategy of no leftovers – the classical cutting stock problem. However, by using $U = 12$, the waste is respectively 118.8 and 5243.5, which means reductions of 13.2% and 24.8%. Considering that in small industries, raw materials have the most important part of the overall cost of final products, this reduction can make them competitive in the market. Despite the fact that waste decreases as demands increase, irregularity of demands is also common for small industries, so that keeping items in stock – the item oriented approach – may be unacceptable. It is also worth noting that small demands for the cutting problem can occur for larger industries, such as the agricultural light aircraft industry in Abuabara and Morabito (2009). In summary, the situations of small to medium demands together with big to medium items are the most promising situations to allow for leftovers during the cutting process.

The following experiments consider an intermediate period, where alongside the standardized objects are also leftovers from previous periods. We started with no leftovers in stock and increased them one by one. Furthermore, the value of the upper bound U varied together with the availability of leftovers to make the stock consistent, but not too tight so that the generation of new leftovers is avoided. That is, for $k=0$ to 3, $e_s=k$, $s=1, 2, 3$, and $U=3k+3$. Note that if $k=0$, then $e_s=0$, $s=1, 2$, and 3, which means that there is no leftover in stock, but there is a possibility of generating $U=3$ new leftovers. This is a real possibility, which is not only for the initial period. Table 2 shows the average waste in 50 instances, and Graph 2 depicts the percentage waste evolution in terms of U related to the classical CSP ($k=0$, $U=0$). The results are quite similar to the first period.

Table 2: Waste for an intermediate period: leftovers in stock

Class	$k = 0, U = 3$	$k = 1, U = 6$	$k = 2, U = 9$	$k = 3, U = 12$
[M,S]	125.1	120.6	118.9	118.8
[M,M]	613.9	598.4	587.4	577.9
[M,B]	3590.3	3582.0	3573.7	3565.6
[B,S]	6343.0	5933.0	5598.5	5359.5
[B,M]	38830.4	38195.7	37572.4	36963.0
[B,B]	210153.7	209435.7	208724.4	208015.0

Graph 2: Percentage waste for an intermediate period related to $k=0, U=0$



It is worth noting that the waste is also dependent on the sizes and availabilities of the leftovers. Let us take an instance of class [B,S] where $e_s=3, s=1, 2, 3$ and $U=12$ (an instance of row [B,S] and column $k=3, U=12$ in Table 2). That is, there are 9 leftovers in stock in total. The result gave a waste of 5124, and the leftovers sized 400 and 500 were fully used, and no leftover sized 600 was used. On the other hand, if the stock of leftovers were 6 units for size 400 and 3 units for size 500, summing 9 as before, the waste would be 4824, which means a waste reduction of 5.6%. This simple example shows that the model (1) – (6) can also be used for lot sizing the leftover stock. Similarly, different sizes of the leftovers can also be tested, that is, the manager can deal with the assortment problem.

5 Conclusions and Final Remarks

In this article, we presented a new mathematical model to describe the practical procedure of allowing partial cutting of objects in stock, creating leftovers to be cut in the future. First versions and discussions of this model were presented in the 9th ESICUP (EURO Special Interest Group on Cutting and Packing) in La Laguna, Spain, 2012, and in the CMAC (Brazilian Congress of Applied Mathematics and Computing), Bauru, Brazil, 2013. This simple and practical mathematical model represents the well-known fact that the diversity of object types in stock makes waste reduction possible. The idea of producing leftovers in order to reduce the waste is well-known in the literature, but only recently have researchers paid more attention to this problem.

For many raw materials, such as steel, after partially cutting an object, the protection against corrosion is lost, and therefore the sooner leftovers are cut the better, otherwise they will be wasted. In such a case, $\alpha'' < 1$ should be used to give priority to cut leftovers. On the other hand, $\alpha' > 1$ can be used to create new leftovers, only if it is really attractive. The model is flexible enough to allow different weights for each leftover, α'_s and α''_s , or even to force some of them to be used, by just including a lower bound in (4).

Although we have fixed the leftover sizes, the manager might decide to add the waste in the cutting pattern to the leftover size. Of course, this makes increasing the number of different leftover types which demands a higher control system.

In addition, model (1) – (6) is quite general and flexible to deal with higher dimensions, as the dimension and additional constraints of the cutting process is embedded in the pattern generation in Step 2 of the Column Generation Algorithm (see Section 3). For example, in cutting plates sized $L \times W$, the manager can decide to have leftovers sized $L_s \times W$, and/or $L \times W_s$, by cutting the current demand using reduced objects sized $(L - L_s) \times W$ and/or $L \times (W - W_s)$. Therefore, in Step 2 of the Column Generation Algorithm, the objective function is subject to $(a_{1s} \dots a_{ms})$ to be a cutting pattern for the plates $(L - L_s) \times W$, or $L \times (W - W_s)$. The way a plate is cut, such as guillotined, 2-staged etc., determines the constraints in the pricing problem.

In order to solve the integer CSLP, heuristics can be designed as extensions of the residual heuristic in Poldi and Arenales (2009). Residual heuristics work as the following: solve LP relaxation of problem (1) – (6) and create rules to round the continuous solution to obtain integer values for the variables. Then, update demands and stocks, and repeat the steps. By solving the pricing problems with upper bounds on the number of items, so that any cutting pattern generated can be used at least once, the residual heuristics always converge.

Acknowledgements

This research has been supported by the *Fundação de Amparo a Pesquisa do Estado de São Paulo* (FAPESP) and the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq).

References

- [1] Abuabara A. and Morabito R. (2009). Cutting optimization of structural tubes to build agricultural light aircrafts. *Annals of Operations Research* 169: 149-165.
- [2] Andrade, R., Birgin, E. G., Morabito, R., and Ronconi, D. P. (2013). MIP models for two-dimensional non-guillotine cutting problems with usable leftovers. *Journal of the Operational Research Society* 65: 1649-1663.

- [3] Andrade, R., Birgin, E. G., Morabito, R. (2014). Two-stage two-dimensional guillotine cutting stock problems with usable leftovers. *International Transactions in Operational Research*, to appear (DOI: 10.1111/itor.12077).
- [4] Brown, A. R. Optimum packing and depletion: the computer in space and resource usage proble. New York: Macdonald - London and American Elsevier Inc, 1971. 107p.
- [5] Cherri A. C., Arenales M. N. and Yanasse H. H. (2009). The one-dimensional cutting stock problems with usable leftovers: A heuristic approach. *European Journal of Operational Research*, 196: 897-908.
- [6] Cherri, A. C., Arenales, M. N., Yanasse, H. H. (2013). The usable leftover one-dimensional cutting stock problem—a priority-in-use heuristic. *International Transactions in Operational Research*, 20: 189–199.
- [7] Cherri, A. C.; Arenales, M. N.; Yanasse, H. H.; Poldi, K. C.; Vianna, A. C. G. (2014).The one-dimensional cutting stock problem with usable leftovers – A survey. *European Journal of Operational Research*, 236: 395-402.
- [8] Cui Y. and Yang Y. (2010). A heuristic for the one-dimensional cutting stock problem with usable leftovers. *European Journal of Operational Research* 204: 245-250.
- [9] Desaulniers, G.; Desrosiers, J.; Solomon, M. M. (Eds). Column Generation. Springer Science and Business Media, Inc., 2005. 358p.
- [10] Gilmore, P.C and Gomory, R.E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research* 9: 849-859.
- [11] Gilmore, P. C., Gomory, R. E. (1963). A linear programming approach to the cutting stock problem - Part II. *Operations Research*, 11: 863-888.
- [12] Gradisar M, Jesenko J and Resinovic C (1997). Optimization of roll cutting in clothing industry. *Computers and Operational Research* 10: 945-953.
- [13] Gradisar M., Kljajic M., Resinovic C. and Jesenko J. (1999). A sequential heuristic procedure for one-dimensional cutting. *European Journal of Operational Research* 114: 557-568.
- [14] Kantorovich, L.V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 6: 366-422.
- [15] Poldi K. C., Arenales M. N. (2009). Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Computers and Operations Research* 36: 2074 – 2081.
- [16] Roodman G. M. (1986). Near-optimal solutions to one-dimensional cutting stock problem. *Computers and Operations Research* 13: 713-719.

- [17] Scheithauer G. (1991). A note on handling residual length. *Optimization*, 22: 461-466.
- [18] Wäscher, G., Gau, T. (1996). Heuristics for the integer one-dimensional cutting stock problem: a computational study. *OR Spektrum*, 18, 131-144.
- [19] Wäscher, G., Haußner, H., Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183:1109-1130.