

See discussions, stats, and author profiles for this publication at: <http://www.researchgate.net/publication/220461904>

An effective solution for a real cutting stock problem in manufacturing plastic rolls.

ARTICLE in ANNALS OF OPERATIONS RESEARCH · JANUARY 2009

Impact Factor: 1.1 · DOI: 10.1007/s10479-008-0407-1 · Source: DBLP

CITATIONS

2

DOWNLOADS

53

VIEWS

78

5 AUTHORS, INCLUDING:



Ramiro Varela Arias

University of Oviedo

66 PUBLICATIONS 283 CITATIONS

SEE PROFILE



Camino Rodríguez Vela

University of Oviedo

62 PUBLICATIONS 287 CITATIONS

SEE PROFILE



Jorge Puente Peinador

University of Oviedo

50 PUBLICATIONS 193 CITATIONS

SEE PROFILE



Ines Gonzalez Rodriguez

Universidad de Cantabria

41 PUBLICATIONS 146 CITATIONS

SEE PROFILE

Annals of Operations Research manuscript No. (will be inserted by the editor)

An effective solution for a real cutting stock problem in manufacturing plastic rolls

Ramiro Varela · Camino R. Vela ·
Jorge Puente · María Sierra · Inés
González-Rodríguez

Received: date / Accepted: date

Abstract We confront a practical cutting stock problem from a production plant of plastic rolls. The problem is a variant of the well-known one dimensional cutting stock, with particular constraints and optimization criteria defined by the experts of the company. We start by giving a problem formulation in which optimization criteria have been considered in linear hierarchy according to expert preferences, and then propose a heuristic solution based on a GRASP algorithm. The generation phase of this algorithm solves a simplified version which is rather similar to the conventional one dimensional cutting stock. To do that, we propose a Sequential Heuristic Randomized Procedure (SHRP). Then in the repairing phase, the solution of the simplified problem is transformed into a solution to the real problem. For experimental study we have chosen a set of problem instances of com-mon use to compare SHRP with another recent approach. Also, we show by means of examples, how our approach works over instances taken from the real production process.

All authors are supported by MEC-FEDER Grant TIN2007-67466-C02-01 and by contract CN-05-127 of the University of Oviedo and the company ERVISA, and by FICYT under grant BP04-021.

Ramiro Varela · Camino R. Vela · Jorge Puente
Computing Technologies Group. Department of Computing, Artificial Intelligence Center,
University of Oviedo, Spain. Campus de Viesques, 33271 Gijón.

Ramiro Varela
E-mail: ramiro@uniovi.es

Camino R. Vela
E-mail: crvela@uniovi.es

Jorge Puente
E-mail: puente@uniovi.es

María Sierra · Inés González-Rodríguez
Department of Mathematics, Statistics and Computing. University of Cantabria, Spain.
Los Castros s/n, 39005 Santander.

María Sierra
E-mail: mariarita.sierra@unican.es

Inés González-Rodríguez
E-mail: ines.gonzalez@unican.es

Keywords Cutting stock · iterative sequential heuristics · randomized algorithms · meta-heuristics · multi-objective optimization.

1 Introduction

In this work we confront a real *cutting stock problem* of a company that manufactures plastic rolls. The process starts by generating big rolls of about 6000 mm. width and a variable length that are then cut into small rolls of the same length whose width ranges from about 300 mm. to 1200 mm. according to customer requirements. Figure 1 shows a schema of the cutting machine. The big roll is set on the cutting machine and unrolled so that the film band passes through the cutter knives and splits into thinner bands that are then rolled into smaller rolls of a given length. The set of rolls generated in this way is called a *cut*, and the configuration of the knives is the *cutting pattern*. Normally, a large number of cuts can be obtained from a big roll, and moreover if the big roll finishes before a cut is completed, a new big roll can be set in the machine and the film joined with the previous roll. Thus the big roll may be supposed to be infinite. In order to ensure the quality of the first and last rolls of the cut, a small amount of film of about 100 mm. should be discarded at each side of the big roll due to plastic defects in the borders, so that there is a *maximum width* of the big roll that is actually useful. Additionally, due to the fixed width of the required rolls, some more plastic should be discarded in each cut. This is the *trim-loss* that should be minimized by the pattern generation procedure. Due to technical characteristics of the machine, there is a maximum trim-loss that the machine is able to manage, so that there is also a *minimum width* that each cut should consume. In practice, these values are fixed to about 5800 mm. and 5500 mm. respectively. During a cut, each roll is supported by a pressure-roller that avoids the generation of folds. The width of a pressure-roller depends on the width of the supported roll, and its weight is of about 30 kg. A pressure-roller of a given width may support rolls in a small range of width. For example a roller of 800 mm. is suitable for rolls from 750 to 790 mm. And for a given roll, only one pressure-roller width serves. The setting of pressure-rollers is made manually by the machine technicians, so minimizing these settings is an important objective, in addition to minimizing the number of cuts and the cost of setups or changeovers. In this work we suppose that the setup cost between two cutting patterns is mainly due to the number of pressure-rollers that have to be put in and taken off, and also due to the difference in the number of cutting knives.

The problem has also a number of constraints and optimization objectives that make it different from the conventional formulations of the one dimensional cutting stock problem given in the literature. For example neither underproduction nor overproduction is allowed for any of the orders, which together with the constraints of maximum and minimum width, makes it difficult, even impossible in some cases, to reach a valid solution. In the last case, the only way to build up an acceptable solution is to produce a number of additional rolls, whose width and number should be explicitly permitted by the expert, to be stored in the shop stock. Once a cut is completed, the rolls are packed into stacks. The stack size is fixed for each roll width, so a given order is composed by a number of stacks, maybe the last one being uncompleted. Naturally, only when a stack is completed is it taken away from the proximity of the cutting machine. So, minimizing the number of open stacks is also convenient in order to facilitate the production process. Moreover, some orders have more priority than

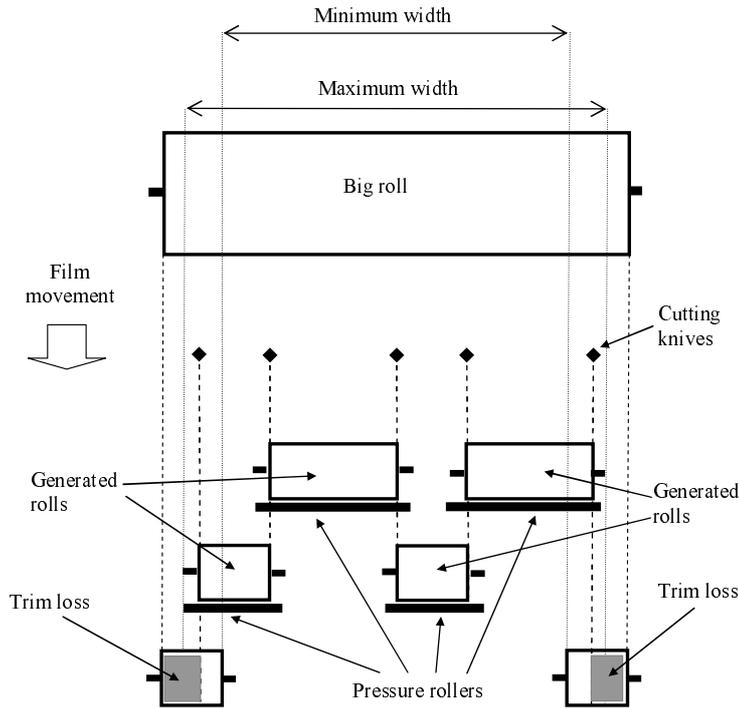


Fig. 1 Working schema of the cutting machine

others. Consequently the delivery time of orders weighted by the client priority is an important criterion as well.

The remainder of the paper is organized as follows. In section 2, we review the classical formulation of the one dimensional cutting stock problem and a number of approaches proposed for this problem over the years. In section 3, we formulate the problem faced herein. First, in section 3.1 we define the main problem that models the actual problem of a company that produces plastic rolls; and then, in section 3.2, we define a simplified problem by relaxing some constraints of the main problem and also simplifying some data. In section 4 we present our approach to the problem: a GRASP algorithm. Section 4.1 describes the generation phase of this algorithm, which consists in a sequential procedure (SHRP) that solves the simplified problem; and section 4.2 describes the repairing phase that consists in transforming a solution for the simplified problem into a solution for the main problem. In section 5, we report results from an experimental study over a benchmark taken from the literature, in order to compare SHRP with another recent procedure with similar characteristics. Finally, in section 6 we summarize the main conclusions of our work and also we propose some ideas for the future

2 Literature Review

In Gilmore and Gomory (1961) and Gilmore and Gomory (1963) the first model is proposed for the *One Dimensional Cutting-Stock Problem*, also denoted 1D-CSP. The problem is defined by the following data: $(m, L, l = (l_1, \dots, l_m), b = (b_1, \dots, b_m))$, where L denotes the length of each stock piece (here the width of the big roll), m denotes the number of piece types (orders) and for each type $i = 1, \dots, m$, l_i is the piece length (roll width), and b_i is the order demand. A *cutting pattern* describes how many items of each type are cut from a stock length. Following Belov and Scheithauer (2006), let \mathbb{Z}_+ denote the set of positive integers and let the column vectors $A^j = (a_{1j}, \dots, a_{mj}) \in \mathbb{Z}_+^m, j = 1, \dots, n$, represent all possible valid cutting patterns, i.e. those satisfying

$$\sum_{i=1}^m a_{ij} l_i \leq L \quad (1)$$

where a_{ij} is the number of pieces of order i that are generated by one application of the cutting pattern A^j . Let $x_j, j = 1, \dots, n$, be the frequencies, i.e. the number of times each pattern is applied in the solution. The model of Gilmore and Gomory aims at minimizing the number of stock pieces, or equivalently minimizing the trim-loss, and is stated as follows

$$\begin{aligned} z^{1D-CSP} &= \min \sum_{j=1}^n x_j \\ \text{s.t.:} \quad &\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, m \\ &x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n. \end{aligned} \quad (2)$$

Even though this model is the most common in the literature, the problem is often formulated in different ways. In Valério de Carvalho (2002) several formulations of the one-dimensional cutting stock are reviewed, and its particular case called the bin-packing problem, and the relations between their relaxations are analysed.

As is pointed in Umetani et al (2003), this problem is often formulated as an integer programming (IP) problem, and its linear programming (LP) relaxation is exploited in many heuristic algorithms; e.g. first solve the LP and then modify the LP solution to an integer solution heuristically. In this approach, however, it is impractical to consider all feasible cutting patterns, which correspond to the columns in an LP formulation. Gilmore and Gomory also proposed a procedure to find out cutting patterns to improve the LP solution by solving the associated knapsack problem. Since then, many approaches have been proposed to optimize the number of cuts that are usually based on branch-and-bound and column generation algorithms, for example the branch-and-cut-and-price algorithm proposed in Belov and Scheithauer (2006). Also, there are heuristic approaches like for example the simulated annealing heuristic proposed in Chen et al (1996). In Gradisar et al (2002) an experimental study of various methods aimed at minimizing trim-loss is presented.

More recently, the cost of factors other than trim-loss, such as the cost of pattern changes or setups or the cost due to open stacks, have become very important as well. Various types of algorithms have been proposed to deal with the cost of pattern changes. Algorithms of the first class are based on pattern generation procedures such

as the one proposed in Suliman (2001). One of the first approaches of this class was a greedy heuristic algorithm, called the sequential heuristic procedure (SHP) proposed in Haessler (1975) and Haessler (1991), that sequentially adds new patterns to the current partial solution until all demands are satisfied. In each step, it generates a number of cutting pattern candidates and selects one with small trim-loss and high frequency. In Vahrenkamp (1996) a variant of SHP is proposed in which a new pattern is calculated by a randomized algorithm. Basic ideas of these and other algorithms are summarized in Haessler (1991). Recently, in Belov and Scheithauer (2007) an adaptive sequential heuristic, termed sequential value correction (SVC), is proposed that aims at minimizing not only the number of setups but also the number of open stacks. However they consider a simplified model in which an order is packed into only one stack, while in real situations it is usual that more than one stack is required for each order. Moreover, they consider a multi-objective search using pareto criterion.

Algorithms of the second type are based on pattern combinations. The first proposals are Johnston (1986) and Goulimis (1990). Both start from a solution obtained by an LP based approach, and then the number of different patterns is reduced by combining two different patterns into one pattern that covers the same amount of product. More recently, in Foerster and Wscher (1999) an algorithm of this type proposed, called KOMBI, which uses many types of combinations; for example three patterns are replaced with two new ones, etc.

The third type is comprised of exact algorithms such as the algorithm proposed in Vanderbeck (2000). This algorithm minimizes the number of different cutting patterns for a given maximum number of stock rolls. It uses a branch-and-bound algorithm in combination with a column generation technique. According to experimental results, it can obtain optimal solutions for many small problems, but fails for several instances of moderate size. In Belov and Scheithauer (2003) a more simple and flexible method is proposed that produces slightly worse solutions in the same amount of time, but it has the capability for reducing the number of setups if the material input is not restricted to the minimum. Moreover, in Belov and Scheithauer (2006) a branch and price schema enhanced with general purpose cutting planes is proposed. With this schema Belov and Scheithauer were able to solve almost all instances of a set with 200 orders each and many instances with 400 orders.

Although the three types commented above are the most classic approaches to setup minimization, there are others, for example, the iterated local search algorithm with adaptive pattern generation proposed in Umetani et al (2003) called ILS-APG. The main components of this algorithm are a procedure to build up an initial solution and a local search method. The local search is first applied to the initial solution, and then to new solutions obtained by randomly perturbing the best solution obtained so far. The neighbourhood of the local search is given by a subset of cutting patterns which have prospect of improving the current solution. As shown in the experimental study reported in Umetani et al (2003), the performance of ILS-APG is comparable to SHP and KOMBI algorithms.

Summarizing, the best current approaches are branch and bound methods such as those proposed in Belov and Scheithauer (2006) and in Vanderbeck (2000). These algorithms can solve exactly many of the instances of practical interest, even though in some cases within a large running time. In spite of that, heuristic methods are also of great interest due to their flexibility and capacity to obtain a variety of solutions in a short time. As pointed in Belov and Scheithauer (2003), these methods result more flexible when optimization criteria other than the number of cuts and the number of patterns

are considered. Moreover the heuristic methods can be used in conjunction with exact algorithms to obtain new upper bounds and to round up non integer solutions, as it is done for example in Belov and Scheithauer (2006).

3 Problem Formulation

As we have pointed out above, our problem is a variant of the 1D-CSP. The classic formulation given in Gilmore and Gomory (1961) and Gilmore and Gomory (1963) is not directly applicable to our case mainly due to the non-overproduction constraint, but it can be easily adapted as we will see in the sequel. We start by giving a detailed formulation of the *main problem*; that considering all characteristics and optimization criteria relevant from the point of view of the experts. As the number of optimization criteria is too large to deal with all of them at the same time, and also as the search space could be very large, we have opted by introducing a *simplified problem*; i.e. a problem with a lower number of objective functions and also with a smaller search space in general. Once the simplified problem is solved, the solution will be adapted to the original problem; in this process all the objectives will be considered. Table 1 summarizes the main notation used from now on to describe the problem data and its solutions.

Table 1: List of notation used across the paper to represent input data and solutions.

$L_{max}, L_{min}, C_{max}$	Maximum and minimum widths, and maximum number of rolls of a cutting pattern.
$W_{max}, W_{min}, \Delta W$	Maximum and minimum widths of a single roll and increment from a valid width to the next one.
SC	Cost magnitude for operations intervening setup
$PR = \{PR_1, \dots, PR_p\}, FPR$	Types of pressure rollers and mapping from allowed roll widths to pressure rollers.
FST	Mapping from allowed roll widths to stack sizes.
$M = \{1, \dots, m\}$	Index set of orders of the main problem.
$b = (b_1, \dots, b_m)$	b_i is the number of rolls of order i .
$l = (l_1, \dots, l_m)$	l_i is the width of the rolls of order i .
$p = (p_1, \dots, p_m)$	p_i is the priority of order i .
$S = \{m + 1, \dots, m + s\}$	Stock types allowed for overproduction.
$bs = (b_{m+1}, \dots, b_{m+s})$	b_{m+i} is the maximum number of rolls of stock type $m + i$ that may be produced.
$ls = (l_{m+1}, \dots, l_{m+s})$	l_{m+i} is the width of rolls of stock type $m + i$.
A	Set of feasible cutting patterns for the main problem.
$A^j = (a_{1j}, \dots, a_{mj}, a_{(m+1)j}, \dots, a_{(m+s)j})$	A feasible cutting pattern, a_{ij} is the number of items of order or stock of type i in pattern A^j .
$L_j, C_j, D_j = L_{max} - L_j$	Total width, number of rolls and trim-loss of pattern A^j .

(Π, x)	A cutting plan for the main problem, i.e. a list of different cutting patterns $\Pi = (A^1, \dots, A^{ \Pi }) \in \mathbf{A}^{ \Pi }$ and their frequencies $x = (x_1, \dots, x_{ \Pi }) \in \mathbb{Z}_+^{ \Pi }$.
s_i	The number of rolls of stock $i \in S$ in the solution.
A_l^j	Cut due to the l th application of pattern A^j according to plan (Π, x) .
$CI(A_l^j)$	The index of a cut in the sequence of cuts defined by a cutting plan (Π, x) .
$CU_{start}(i), CU_{end}(i)$	Indexes of the first and last cuts of order i in the cut sequence.
$SU(A^{j-1}, A^j)$	The setup cost from pattern A^{j-1} to pattern A^j .
$R(i, A_l^j)$	The number of rolls of order i generated from the beginning up to completion of cut A_l^j .
$OS(i, A_l^j)$	Logical variable indicating if after cut A_l^j there is an open stack of order i .
$L = \{l_e, e \in M \cup S\}, L = ms$	Set of different widths of orders and stock types in the main problem, ms denotes its cardinal.
$M' = \{1, \dots, m'\}$	Index set of orders of the simplified problem.
$b' = (b'_1, \dots, b'_{m'})$	b'_i is the number of rolls of order i in the simplified problem.
$l' = (l'_1, \dots, l'_{m'})$	l'_i is the width of the rolls of order i in the simplified problem.
$S' = \{m' + 1, \dots, m' + s\}$	Stock types allowed for overproduction, type $m' + i$ being the same as type $m + i$ of S .
$bs' = (b'_{m'+1}, \dots, b'_{m'+s})$	$b'_{m'+i}$ is the maximum number of rolls of stock type $m' + i$ that may be produced.
$ls' = (l'_{m'+1}, \dots, l'_{m'+s})$	$l'_{m'+i}$ is the width of rolls of stock type $m' + i$ of the simplified problem, they are different from each other, but it may happen that $l' \cap ls' \neq \emptyset$.
E	Set of feasible cutting patterns for the simplified problem.
$E^j = (e_{1j}, \dots, e_{msj})$	A feasible cutting pattern for the simplified problem, e_{ij} is the number of items of either order or stock of type i in pattern E^j .
(Π', x')	A cutting plan for the simplified problem, i.e. a list of different cutting patterns $\Pi' = (E^1, \dots, E^{ \Pi' }) \in \mathbf{E}^{ \Pi' }$ and their frequencies $x' = (x'_1, \dots, x'_{ \Pi' }) \in \mathbb{Z}_+^{ \Pi' }$.
s'_i	The number of rolls of stock $i \in S'$ in the simplified solution.

3.1 The Main Problem

In order to clarify the problem definition, we present the data of the machine environment and the clients' orders, the form and semantics of a problem solution, the problem constraints and the optimization criteria in the hierarchical order usually considered by the expert.

Data

- The set of parameters of the cutting machine: the maximum width of a cut L_{max} , the minimum width of a cut L_{min} , the maximum number of rolls that can be generated in a cut C_{max} , the minimum and the maximum width of a single roll, W_{min} and W_{max} respectively, and the increment of width ΔW between two consecutive permitted roll widths.
- The setup costs. There is an elementary setup cost SC and some rules given by the expert that allows calculating the total setup cost from a configuration of the cutting machine to the next one. The setup cost is due to roller and cutter changes as follows. The cost of putting in or taking off a pressure-roller is SC . The cost of putting in an additional knife is $3SC$ and the cost of dismounting a knife is $2SC$, as this operation takes a lower time. Here is important to remark that the costs of displacing an installed knife to other position has a null cost due to this process is done by the machine without technicians intervention. Also, moving a roller inside the machine is considered to have null cost, even though this process is done by a technician. Of course, this is a simplified view of the actual setup cost and is founded on agreement of technicians with experts.
- The types of pressure-rollers $PR = \{PR_1, \dots, PR_p\}$ and the mapping F_{PR} from roll widths to pressure-rollers.
- The mapping F_{ST} from roll widths to stack sizes or number of rolls in each stack unit.
- The orders description given by $(M = \{1, \dots, m\}, b = (b_1, \dots, b_m), l = (l_1, \dots, l_m), p = (p_1, \dots, p_m))$ where for each order $i = 1, \dots, m$, b_i denotes the number of rolls, l_i denotes the width of the rolls and p_i the order priority.
- The stock allowed for overproduction $(S = \{m + 1, \dots, m + s\}, bs = (b_{m+1}, \dots, b_{m+s}), ls = (l_{m+1}, \dots, l_{m+s}))$ where for each $i = 1, \dots, s$, b_{m+i} denotes the number of rolls of type $m + i$ allowed for overproduction and l_{m+i} denotes the width of these rolls.
- The set of feasible cutting patterns, for the orders and stock given, \mathbf{A} where each $A^j \in \mathbf{A}$ is, $A^j = (a_{1j}, \dots, a_{mj}, a_{(m+1)j}, \dots, a_{(m+s)j}) \in \mathbb{Z}_+^{m+s}$ and denotes that, for each $i = 1, \dots, m + s$, a_{ij} rolls of order i are cut each time the cutting pattern A^j is applied. A cutting pattern A^j is feasible if and only if both of the following conditions hold

$$L_{min} \leq L_j = \sum_{i \in M \cup S} a_{ij} l_i \leq L_{max} \quad (3)$$

$$C_j = \sum_{i \in M \cup S} a_{ij} \leq C_{max} \quad (4)$$

where L_j and C_j are the total width and the number of rolls of pattern A^j respectively. $D_j = L_{max} - L_j$ de-notes the trim-loss of the cutting pattern. Moreover, we

only consider proper cutting patterns, i.e. patterns with $a_{ij} \leq b_i, i = 1, \dots, m + s$, $A^j \in \mathbf{A}$.

Goal

The objective is to obtain a *cutting plan* (Π, x) , where $\Pi = (A^1, \dots, A^{|\Pi|}) \in \mathbf{A}^{|\Pi|}$ and $x = (x_1, \dots, x_{|\Pi|}) \in \mathbb{Z}_+^{|\Pi|}$ denotes the pattern frequencies. The cutting patterns of Π are applied sequentially, each one the number of times indicated by its frequency. $A_l^j, 0 \leq j \leq |\Pi|, 0 \leq l \leq x_j$, denotes the l th cut corresponding to pattern A^j and $CI(A_l^j)$ is the cut index defined as

$$CI(A_l^j) = \sum_{k=1}^{j-1} x_k + l. \quad (5)$$

Given an order $i \in M$ its first roll is generated in cut A_1^j such that A^j is the first pattern of Π with $a_{ij} \neq 0$, this cut is denoted $CU_{start}(i)$. Analogously, the last roll of order i is generated in cut $A_{x_k}^k$ so that A^k is the last pattern of Π with $a_{ik} \neq 0$, this cut is denoted $CU_{end}(i)$.

Constraints

As we have considered feasible cutting patterns, the only constraint that should be required for a solution is the following

- The set of rolls generated by the application of the cutting plan (Π, x) should be composed of all rolls from the orders and, eventually, of a number of rolls from the stock. That is, let s_i be the number of rolls of stock $i \in S$ in the solution

$$\forall i \in S, \quad s_i = \sum_{A^j \in \Pi} a_{ij} x_j. \quad (6)$$

Then, the constraint can be expressed as follows:

$$\forall i \in M, \quad \sum_{A^j \in \Pi} a_{ij} x_j = b_i, \quad (7)$$

$$\forall i \in S, \quad 0 \leq s_i \leq b_i. \quad (8)$$

Optimization criteria

Regarding the objective functions, as we have remarked, we consider a number of them in hierarchical ordering according to the expert preferences. Therefore, we will try to optimize the first one, in the case of a tie we will try to optimize the second one, and so on. The objective functions are the following.

1. Minimize the number of cuts, given by $\sum_{j=1}^{|\Pi|} x_j$. The optimum value is denoted z^{1D-CSP} .
2. Minimize the setup cost, given by $\sum_{j=1}^{|\Pi|} SU(A^{j-1}, A^j)$, where $SU(A^{j-1}, A^j)$ denotes the setup cost from pattern A^{j-1} to pattern A^j calculated as it was indicated above in the data section. Configuration A^0 refers to the situation of the cutting machine previous to the first cut.

3. Maximize the amount of stock generated, $\sum_{i \in S} l_i s_i$, in this way the actual trim-loss is minimized for a given number of cuts.
4. Minimize the completion times of orders weighted by their priorities given by

$$\sum_{i \in M} p_i CI(CU_{end}(i)). \quad (9)$$

5. Minimize the maximum number of open stacks along the cut sequence. Let $R(i, A_l^j)$ denote the number of rolls of order i generated from the beginning up to the completion of cut A_l^j

$$R(i, A_l^j) = \sum_{k=1}^{j-1} a_{ik} x_k + a_{ij} l, \quad (10)$$

and let $OS(i, A_l^j)$ be 1 if after cut A_l^j there is an open stack of order i and 0 otherwise, i.e.

$$OS(i, A_l^j) = \begin{cases} 1, & \text{if } \left((R(i, A_l^j) \bmod F_{ST}(l_i)) > 0 \right) \text{ and } (A_l^j \neq CU_{end}(i)), \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Then, the maximum number of open stacks along the cut sequence is given by

$$\max_{\substack{j=1, \dots, |\Pi| \\ l=0, \dots, x_j}} \sum_{i=M} OS(i, A_l^j). \quad (12)$$

3.2 The Simplified Problem

In the main problem, as formulated in previous section, it is often the case that two or more orders have the same width or a stock has the same width as one of the orders. So, from the point of view of the cutting process, two cutting patterns A^i and A^j are equivalent if both patterns define the cutting of the same number of rolls of the same sizes, i.e. given the set of widths $L = \{l_e, e \in M \cup S\}$, with cardinality $|L| = ms$, $ms \leq m + s$.

$$A^i \equiv A^j \Leftrightarrow \sum_{\substack{k=0 \\ l_k=l}}^{m+s} a_{ki} = \sum_{\substack{k=0 \\ l_k=l}}^{m+s} a_{kj}, \quad \forall l \in L \quad (13)$$

Now the simplified problem can be stated as follows

Data

- The set of parameters of the cutting machine: as it is in the main problem.
- The setup costs: as they are in the main problem.
- The types of pressure-rollers and mapping F_{PR} : as they are in the main problem.
- The mapping function F_{ST} : as it is in the main problem.

- The simplified orders description given by $(M' = \{1, \dots, m'\}, b' = (b'_1, \dots, b'_{m'}), l' = (l'_1, \dots, l'_{m'}))$, where for each order $i = 1, \dots, m'$, b'_i denotes the number of rolls and $l'_i \in L$ denotes the width of the rolls. The simplified orders list b' are obtained from the original order list b so as

$$b'_i = \sum_{\substack{k=1 \\ l_k=l'_i}}^m b_k \quad (14)$$

Here it is important to remark that all orders are available at the time of computing a cutting plan, i.e. dynamic planning is not considered.

- The stock allowed for overproduction $(S' = \{m' + 1, \dots, m' + s\}, bs' = (b'_{m'+1}, \dots, b'_{m'+s}), l'_s = (l'_{m'+1}, \dots, l'_{m'+s}))$ where for each $i = 1, \dots, s$, $b'_{m'+i} = b_{m+i}$ denotes the number of rolls of type $m' + i$ allowed for overproduction and $l'_{m'+i} = l_{m+i} \in L$ denotes the width of these rolls (notice that two different stock orders cannot have the same width). Here both l' and ls' are lists with no repeated elements, so they can be seen as sets such that $l' \cup ls' = L$, although, it is possible that $l' \cap ls' \neq \emptyset$. In what follows, we assume L to be ordered, beginning with $l'_1, \dots, l'_{m'}$ followed by the elements from ls' that do not belong to l' . $L = (l'_1, \dots, l'_{ms})$, $ms \leq m' + s$.
- The set of simplified feasible cutting patterns for the simplified orders and stock given, \mathbf{E} , obtained from the set of feasible cutting patterns for the original problem \mathbf{A} , $|\mathbf{E}| \leq |\mathbf{A}|$, where every $E^j \in \mathbf{E}$ is $E^j = (e_{1j}, \dots, e_{msj}) \in \mathbb{Z}_+^{ms}$ meaning that, for each $i = 1, \dots, ms$, e_{ij} rolls of width l'_i are cut each time the cutting pattern E^j is applied. In other words, each element of \mathbf{E} is an equivalence class of the quotient set of \mathbf{A} with the above relation, so it is a simplified representation of a number of cutting patterns of \mathbf{A} .

Goal

The objective is to obtain a *simplified cutting plan* (Π', x') , where $\Pi' = (E^1, \dots, E^{|\Pi'|}) \in \mathbf{E}^{|\Pi'|}$ and $x' = (x'_1, \dots, x'_{|\Pi'|}) \in \mathbb{Z}_+^{|\Pi'|}$ denotes the pattern frequencies.

Constraints

As all the simplified cutting patterns are feasible, the only additional constraint that should be required to a solution is the following

- The set of rolls generated by the application of the simplified cutting plan (Π', x') should be composed of all rolls from the orders and, eventually, of a number of rolls from the stock. That is, let s'_i the number of rolls of stock of width l'_i in the solution, being 0 if there is no $m' + k \in S'$ such that $l'_i = l'_{m'+k}$,

$$\forall i \in \{m' + 1, \dots, ms\}, \quad s'_i = \sum_{E^j \in \Pi'} e_{ij} x'_j \quad (15)$$

Then, the constraint can be expressed as follows:

$$\forall i \in M', \quad \sum_{E^j \in \Pi'} e_{ij} x'_j = b'_i + s'_i, \quad (16)$$

$$0 \leq s'_i \leq b'_{m'+k} \quad (17)$$

Optimization criteria

1. Minimize the number of cuts calculated by $\sum_{j=1}^{|II'|} x'_j$.
2. Minimize the number of simplified cutting patterns $|II'|$.
3. Maximize the amount of stock generated, that is $\sum_{i=1}^{ms} l'_i s'_i$, so the trim-loss is minimized for a given number of cuts.

Now let us clarify how a solution of the simplified problem can be transformed into a solution to the main problem. To do so, we have to map each simple cut from a simplified pattern E^j to any of the cuts of pattern A^k of the equivalence class defined by E^j . In doing so, we can consider different orderings in the simplified cutting plan, and also different orderings between the single cuts derived from a simplified cutting pattern, in order to satisfy all the optimization criteria of the main problem. As we can observe, objectives 1 and 3 are the same in both problems, but objective 2 is different. The reason to consider objective 2 in the simplified problem is that in minimizing the number of patterns $|II'|$ it is expected that the setup cost of the main problem is to be minimized as well. This is because the setup cost between two consecutive cuts A^k and A^l of the main problem is null if both A^k and A^l belongs to the same equivalence class E^j .

4 A GRASP Algorithm for the Cutting Stock Problem

According to the problem formulation, we have developed a procedure to solve it that works in two steps. First, a set of solutions is calculated for the simplified problem; then a selection of these solutions is transformed into solutions to the main problem, as shown in Algorithm 1. Here we describe this algorithm as a GRASP (Greedy Randomized Adaptive Search Procedure), even though it differs from the general structure of GRASP described in Resende and Ribeiro (2002). The main difference is that in a general GRASP, once a solution is obtained by a stochastic method, this solution is repaired by a local search algorithm before obtaining the next stochastic solution, and this process is iterated a number of times. In our approach, all the stochastic solutions are first calculated and only a selection of them is passed to the repairing algorithm. We proceed in this way in order to save time. As the repairing algorithm will never reduce the number of cuts of a solution, only those solutions of the simplified problem with the lowest number of cuts are considered for repairing. Moreover, the repairing mechanism may be implemented by strategies other than local search, such as greedy or even an evolutionary algorithm.

Input An instance (m, L, l, b, S) of a 1D-CSP

Output A set of feasible solutions II

Calculate a set of solutions II' for the simplified 1D-CSP (Algorithm 2);

Repair the simplified solutions II' to obtain a set of solutions II to the main problem (Algorithm 3);

Algorithm 1: GRASP Algorithm for the Main 1D-CSP

4.1 Solving the simplified problem

The simplified problem is solved by means of a Sequential Heuristic Randomized Procedure (SHRP) very similar to the Sequential Value Correction (SVC) proposed in Belov and Scheithauer (2007). These procedures construct a solution pattern after pattern and assign a frequency to each one. Various solutions are calculated iteratively and, in each iteration, some information from previous solutions is used. In principle, to obtain the next pattern to be included in a solution, we consider the residual problem and search for a feasible pattern having the minimum trim-loss and the maximum frequency. However, this strategy does not lead to a good solution in most cases. It often happens that the last patterns have a large trim-loss because the first ones consume those rolls that can be combined easily to obtain good patterns. So, in general, it is better not to select the best patterns at the beginning, in order to have the chance to obtain good patterns at the end as well. In order to do so, in Belov and Scheithauer (2007) the SVC procedure is proposed, which is based on the concept of pseudo-price. The idea is quite simple; at the beginning each roll has a price equal to its width l_i . The algorithm constructs sequentially a number of solutions, where each solution is constructed pattern after pattern. In each step, the pattern with the largest value is calculated, where the value of a pattern is obtained by adding the values of its rolls. In the subsequent iterations the price (or pseudo-price) of the rolls is updated in the following way. Each time a pattern is calculated, the prices of the rolls contained in the pattern are augmented in direct ratio with the trim-loss of the pattern. In this way those rolls that are difficult to combine gradually get a higher price so that, after a number of iterations, they have the chance of being combined at the beginning, giving rise to acceptable patterns.

Our approach differs from that of Belov and Scheithauer mainly in the following aspects. Firstly, we do not calculate the optimal pattern in each step, but use a heuristic algorithm that produces a sub-optimal pattern instead. Also, we do not modify the price of the rolls. Moreover, we use some heuristic strategies aimed at facilitating the generation of good patterns at the end. One of these heuristics consists in penalizing the number of rolls when the price or value of a pattern is calculated. In this way we expect to have, at the beginning, patterns with few rolls of large width, thus leaving for the last patterns many of the smaller rolls that are expected to be easier to combine. Hence, the value of a pattern E^j is calculated by

$$\text{VALUE}(E^j, p) = C^p \sum_{\substack{i=1 \\ b_i'' > 0}}^{m'} e_{ij} l_i' \quad (18)$$

where C is the number of rolls of the pattern, p is a parameter and b_i'' is the number of rolls of order i in the residual problem. In this way those rolls of pattern E_j that are taken from the stock S' , and consequently do not correspond to any of the orders, do not contribute to the value of the pattern.

As it can be expected, the best value of p depends on the problem structure and is not easy to envisage a priori. Hence, for the first solutions the value of p is selected before starting to construct a cutting plan in direct ratio with the square root of $|p|$, from an interval $[-P, 0]$, with $1 \geq P > 0$. Then, in subsequent solutions those values that have produced the best solutions will be given a larger probability of being selected.

When a cutting pattern must be obtained in the process of building a solution, we propose a heuristic algorithm that searches for a pattern with both a large value and a large frequency. Furthermore, we will try to consume the rolls uniformly in order to avoid situations where many rolls of the same order remain unpacked when constructing the last patterns. If this were the case, these rolls might be difficult to combine. To avoid this situation, we consider the residual problem b'' and generate a suitable subset of feasible patterns $SE \subset \mathbf{E}$; then we choose the pattern of SE with the largest value, breaking ties using pattern frequency.

The maximum cardinality of set SE is fixed before calculating each pattern. First, a fixed value that depends on both the value of a parameter SE_{min} and the size of the residual problem is given. Then, a random value which is controlled by parameter SE_{gap} is added. Hence, it is calculated as

$$|SE| = 0.1 * SE_{min} + 0.9 * (S_{RP}/S_{OP}) * SE_{min} + \text{Rand}(0, SE_{gap}), \quad (19)$$

where S_{OP} and S_{RP} represent the size of the original problem and the size of the residual problem respectively calculated as

$$S_{OP} = \sum_{i=1}^{m'} b_i' l_i', \quad (20)$$

$$S_{RP} = \sum_{i=1}^{m'} b_i'' l_i''. \quad (21)$$

The construction of a cutting plan is aborted if the sum of a lower bound of the number of cuts of the residual problem, $LB(b'')$, plus the actual number of cuts reaches the current upper bound. Here we have used a trivial lower bound calculated as S_{RP}/L .

The scheme of the whole method is given in Algorithm 2. The function $\text{Rand}(r_1, r_2)$ returns, in principle, real numbers uniformly in the interval $[r_1, r_2]$, but as long as improving solutions appear, the probability distribution is modified so as the values that have produced the last improvements get a larger probability in the short term. In practice these values are given probability 1 for the next five consecutive times. The set of solutions Π is updated so that at the end it contains the solutions with the minimum number of cuts. These are the solutions that will be transformed into actual solutions to the original problem by the repairing procedure.

4.2 Repairing simplified solutions

The repairing mechanism that produces solutions to the main problem from solutions to the simplified problem depends on the expert criteria about the evaluation functions. In any case, the first criterion, that is the number of cuts, is the most important; but there could be differences in the remaining functions. Here, we have assumed a hierarchy among the five objectives as they are declared in Section 3.1. The number of cuts and the amount of stock generated are fixed in the simplified solutions and in any case cannot be modified by the repairing procedure. On the other hand, the value of the remaining evaluation functions is determined by the repairing method.

Input An instance (m', L, l', b') of a 1D-CSP,
A set of parameters $\{P, SE_{min}, SE_{gap}, IterationsLimit\}$
Output A set of feasible solutions Π'
Initialize $\Pi' = 0; UB = \infty;$
repeat
 $b'' = b'; \pi = 0; x = 0; p = \text{Rand}(-P, 0);$
repeat
 $SE = \text{subset of random patterns of } \mathbf{E} \text{ suitable for } b'';$
 $E = \arg \max\{\text{VALUE}(E, p), E \in SE\};$
 $x_E = \min_{i: e_i > 0} \lfloor b''_i / e_i \rfloor;$
 $x = x + x_E; \pi = \pi + E;$
 $b'' = b'' - x_E E;$
until $(b''_i = 0, \forall 1 \leq i \leq m') \text{ or } (\sum_i x_i + LB(b'') < UB);$
if (π, x) is a solution **then**
if $\sum_i x_i \leq UB$ **then**
update UB and Π' with $(\pi, x);$
if (π, x) is an improving solution **then**
augment the probability of value $p;$
until $IterationsLimit$ is exceeded;

Algorithm 2: Sequential Heuristic Randomized Procedure for the simplified problem

Given a solution of the simplified problem (Π', x') , where $\Pi' = (E^1, \dots, E^{|\Pi'|}) \in \mathbf{E}^{|\Pi'|}$ and $x' = (x'_1, \dots, x'_{|\Pi'|}) \in \mathbb{Z}_+^{|\Pi'|}$, with a number of cuts

$$z = \sum_{i=1}^{|\Pi'|} x'_i, \quad (22)$$

we first consider the equivalent extended solution, that containing each pattern as many times as its frequency in the simplified solution, denoted as $\Pi'_{ext} = (E^1_1, \dots, E^1_{x'_1}, \dots, E^{|\Pi'|}_1, \dots, E^{|\Pi'|}_{x'_{|\Pi'|}}) = (E^1, \dots, E^z)$. This extended solution is transformed into a solution to the main problem. In this process the rolls of the simplified cutting patterns are assigned to orders of the main problem according to their priorities, leaving the last rolls for stock. This aims at minimizing the weighted completion times. In the sequel, we explain this process more formally. Given a simplified pattern $E^j = (e_{1j}, \dots, e_{msj})$ of Π'_{ext} , e_{ij} $1 \leq i \leq ms$ denotes a simplified cut whose width is l'_i . If $1 \leq i \leq m'$, the rolls correspond to order i and, maybe, to the stock and if $m' + 1 \leq i \leq ms$ they are just stock rolls. Let e_i be the ordered list of simplified cuts with width l'_i , i.e. $e_i = (e_{i1}, \dots, e_{iz})$. Now let $k_i = (k_1, \dots, k_l)$ be the orders of the main problem whose width is l'_i , sorted by their priorities, i.e. $p_{k_1} \geq \dots \geq p_{k_l}$, and let $m + i', i' \leq s$, be the stock type of the main problem with the same width, i.e. $l_{k_1} = \dots = l_{k_l} = l_{m+i'} = l'_i$. The rolls from the simplified list of cuts e_i are distributed among the orders from k_i and the stock $m + i'$ so that the following two conditions hold:

- For all $k', k'' \in k_i$ with $k' < k''$ and for all $e_a, e_b \in e_i$ with $e_a < e_b$, if order k' has a roll from e_b , then order k'' has none of the rolls from e_a .
- For all $k \in k_i$ and for all $e_a, e_b \in e_i$ with $e_a < e_b$, if order k has a roll from e_b , then the stock $m + i'$ has none of the rolls from e_a .

In doing so, each simplified cut e_{ij} of E^j is transformed in a set of actual cuts $a_{i_1j}, \dots, a_{i_lj}, a_{(m+i')j}$. Therefore, each instance of E^j in Π'_{ext} is transformed in an

actual pattern, i.e. an element of the equivalence class E^j . And, finally, Π'_{ext} is transformed in an extended plan $\Pi_{ext} = (A_1^1, \dots, A_{x_1}^1, \dots, A_1^{|\pi'|}, \dots, A_{x_{|\Pi'|}}^{|\Pi'|})$. The patterns of Π_{ext} can be sorted in order to improve the changeovers or the number of open stacks, maybe at the cost of worsening the weighted completion time. After that, Π_{ext} becomes a sequence of patterns of the form $(A_1^1, \dots, A_{x_1}^1, \dots, A_1^i, \dots, A_{x_i}^i, \dots, A_1^n, \dots, A_{x_n}^n)$, $(A_1^i, \dots, A_{x_i}^i)$ being a sequence of x_i instances of the same pattern A^i . Hence, we finally obtain the cutting plan given by (Π, x) , where $\Pi = (A^1, \dots, A^n)$ and $x = (x_1, \dots, x_n)$, with $n = |\Pi| \geq |\Pi'|$.

Input A solution (Π', x') of the simplified problem

Output A solution (Π, x) of the main problem

Order Π' , and x' , in increasing, or decreasing, order of number of rolls by pattern;

Π'_{ext} = Extended solution of the simplified problem obtained from Π' ;

Π_{ext} = Extended solution of the main problem obtained from Π'_{ext} ;

(Π, x) = Actual solution to the main problem obtained from Π_{ext} ;

Algorithm 3: Transforming a solution to the simplified problem into a solution of the main problem

Let us clarify how Algorithm 3 works by means of a simple example. The problem data and final results are displayed similarly as they are by the developed application. Table 2 shows an instance and the corresponding simplified problem. A real instance is given by a set of orders, each one defined by a client name, a client identification number, the number of rolls (#Rolls), the width of the rolls and the order priorities(OP). Additionally, the maximum and minimum allowed width of a cut should be given, in this case 5500 and 5700 respectively and also a stock description to choose a number of rolls from if it is necessary to obtain valid cutting patterns. In this example up to 10 rolls of each width 1100, 450 and 1150 could be included in the cutting plan. Furthermore, some other parameters (not shown in Figures) are necessary, for instance, two additional data items should be given to evaluate the number of open stacks and setup cost: the number of rolls that fit in a stack (mapping F_{ST}) and the correspondence between the size of pressure rollers and the width of the supported rolls (mapping F_{PR}). Here we have supposed that every stack contains 4 rolls and that the correspondence between pressure roller types and width rolls is the following: type 1 (0-645), type 2 (650-1045), type 3 (1050-1345), type 4 (1350-1695). All the allowed widths are multiples of 5 and the minimum width of a roll is 250 while the maximum is 1500. Finally, the maximum number of rolls in a pattern is 10.

As we can observe in Table 2, the main instance with 10 orders is reduced to a simplified instance with only 6 orders. This simplified instance is actually a conventional 1D-CSP instance with two additional constraints: the maximum number of rolls in a pattern and the minimum width of a pattern. Table 3 shows a solution to the simplified problem with 21 cuts and 4 different patterns, where 3 stock rolls have been included in order that the last pattern to be valid.

Table 4 shows the final solution to the main problem obtained from the simplified solution of Table 3 by application of Algorithm 3. The upper part of this figure shows the cutting plan. As we can observe each pattern shows the roll widths and there are various patterns with the same roll widths. These patterns are not actually the same as the rolls correspond to different orders. The middle part of Table 4 shows the order

Table 2 An example of problem data

Main Problem					Stock		Simplified Problem	
#Rolls	Width	Order	Client	OP	#Rolls	Width	#Rolls	Width
20	600	20001	Client 1	1	10	1500	30	600
10	600	20002	Client 2	2	10	550	28	850
15	850	20003	Client 3	1	10	1150	15	950
13	850	20004	Client 4	1			14	1350
15	950	20005	Client 5	1			20	550
14	1350	20006	Client 6	1			33	900
20	550	20007	Client 7	1				
18	900	20008	Client 8	1				
15	900	20009	Client 9	2				

Table 3 A cutting plan for the simplified instance of Table 2

FREQUENCY	14	3	3	1
PATTERNS	600	550	950	900
	900	950	950	600
	1350	950	900	600
	850	550	900	1500
	600	900	950	1500
	850	900	900	550
	550	900		
PATTERN WIDTH	5700	5700	5550	5650
OBJECTIVE FUNCTIONS				
Trim loss				500
	0	0	150	50
Number of Patterns				4
Number of Cuts				21

identifiers, where 0 represents to the stock. Here all patterns are different. The bottom of Table 4 shows the values of the evaluation functions. The changeover of each pattern refers to the cost of put in and out cutting knives and pressure rollers from the previous pattern to the current one. As we can observe the first pattern has a changeover cost of 24 because it is assumed that it is necessary to put in all the cutting knives and pressure rollers before this pattern. In practice this is not often the case as a number of cutting knives and pressure rollers remain in the machine from previous cuts. Regarding open stacks, each column of Table 4 shows the number of them that remain incomplete in the proximity of the machine from a cut to the next one, i.e. when a stack gets full after a cut, or it is the last stack of an order, it is not considered.

To obtain the solution of Table 4 from the simplified solution of Table 3, the first step is to assign an order identifier to each roll. To do so, the simplified cuts are considered in the order derived from the simplified solution, i.e. 14 cuts of the first pattern followed by 3 cuts of the second, and so on. Then, each roll is assigned to the available order with the largest priority. For example, the rolls of width 600 of the first 4 cuts are assigned to order 20002 which has a larger priority than order 20001. In

Table 4 Final solution to the main problem of Table 2

FREQUENCY							
1	4	2	1	7	2	1	3
PATTERNS							
900	600	600	600	600	550	550	950
600	900	900	900	900	950	950	950
600	1350	1350	1350	1350	950	950	900
1500	850	850	850	850	550	550	900
1500	600	600	600	600	900	900	950
550	850	850	850	850	900	900	900
	550	550	550	550	900	900	
PATTERN WIDTH							
5650	5700	5700	5700	5700	5700	5700	5550
ORDER IDENTIFIERS							
20009	20002	20001	20001	20001	20007	20007	20005
20002	20009	20009	20009	20009	20005	20005	20005
20002	20006	20006	20006	20006	20005	20005	20008
0	20004	20004	20004	20003	20007	0	20008
0	20002	20001	20001	20001	20008	20008	20005
20007	20004	20004	20003	20003	20008	20008	20008
20007	20007	20007	20007	20008	20008		
OBJECTIVE FUNCTIONS							
Trim Loss							500
50	0	0	0	0	0	0	150
Changeovers							41
24	5	0	0	0	0	4	5
Open Stacks							55
3	4-4-2-3	5-3	2	3-4-4-3-2-4-1	3-1	2	1-1-0
Weighted Time							934
Num. of Cuts							21

this way, the pattern with frequency 14 in the simplified solution is transformed into four patterns with frequencies 4, 2, 1 and 7 respectively in the final solution (columns 2-5 in Table 4). Then, the patterns are sorted by a greedy algorithm that tries to minimize changeovers. This algorithm starts by selecting one of the patterns with the largest number of rolls; then, it tries each one of the remaining patterns just after and before the partial plan. The pattern yielding lowest changeover is selected in the corresponding position (i.e. before or after the previously selected patterns), and so on. In the case of tie, the pattern with the largest priority (given by summing up the rolls' priorities) is selected to appear before. As the open stacks cannot be estimated until the cutting plan is completed, this criterion is only used, in principle, as the last one in the hierarchy of the objective functions for the purpose of comparing solutions.

5 Experimental Study

In this section we describe the experimental study that we have done to test the Sequential Heuristic Randomized Procedure (SHRP) and to compare it with other recent approaches. To do this, we have chosen the Sequential Value Correction (SVC) procedure and the benchmark problems proposed in Belov and Scheithauer (2007). These problems can be downloaded from <http://www.math.tu-dresden.de/~capad/>. The set contains 9 classes of problems (m, L, l, b) with 100 instances each; with parameters $L = 10000$, $m \in \{20, 50, 150\}$, $l_i \in [v_1 L, v_2 L] \cap \mathbb{Z}$ uniformly distributed with $v = (v_1, v_2) \in \{v^1 = (0.01, 0.02), v^2 = (0.01, 0.4), v^3 = (0.01, 0.7), v^4 = (0.2, 0.4), v^5 = (0.2, 0.7)\}$; and $b_j \in S_j$ uniformly distributed with $S_1 = [1, 10] \cap \mathbb{Z}$, $S_2 = [1, 100] \cap \mathbb{Z}$, $S_3 = [50, 100] \cap \mathbb{Z}$. The basic class was $(m = 50, v = v^3 = (0.01, 0.7), b_i \in S_3)$. By varying one of the parameters, all 9 classes are obtained. In the first set of experiments, we have considered only the classes 1, 4 and 9 (as was done in Belov and Scheithauer (2007) for some preliminary experiments) that correspond to the parameters $(m = 20, v = v^3, b_i \in S_3)$, $(m = 50, v = v^4, b_i \in S_3)$ and $(m = 150, v = v^3, b_i \in S_3)$ respectively.

In order to compare SHRP with SVC over the previous instances, we have to do some simplifications in the SHRP strategy. First, the value of L_{min} is set to 0, because SVC searches for patterns without this restriction. Also, there is no stock available because SVC does not consider this possibility either. In spite of these simplifications, there is still an important difference among these methods. SVC aims at minimizing the number of cuts only, while SHRP aims at minimizing the number of patterns too.

Table 5 summarizes the results of SVC across classes 1, 4 and 9, as they are reported in Belov and Scheithauer (2007). In this experimental study, Belov and Scheithauer consider different values of a parameter ρ that controls the value or pseudo-price of a pattern. They experiment with values of $\rho \in \{1.005, 1.01, 1.02, 1.03, 1.04\}$, and run the algorithm up to 200 iterations for each value of ρ . Also, they experiment with a random choice of ρ according to a uniform distribution from the interval $[1.01, 1.03)$. The computer was Athlon 1400 MHz with 512 MB of RAM. As can be observed in Table 5, SVC is quite sensitive to the value of ρ . For each class of problems, there is a value that seems to be the best one, and this value differs from one problem to another. Moreover, a random selection over an interval is good in some cases, as for class 1, but it is clearly worse in other cases, as in class 9. Similar results are reported in Belov and Scheithauer (2007) for the remaining classes. In any case, with random selection the SVC does not reach much better solutions even if it is run for a larger number of iterations.

We have conducted similar experiments with SHRP on a computer Pentium IV 1700 MHz with 512 of RAM. In order to compare with SVC, we have considered that our cpu is 1.16 times faster than Athlon 1400, in accordance with some experimental results from the CASE laboratory (<http://www.caselab.okstate.edu>). Hence, we parameterized SHRP so as to run for a similar period of time. From a set of preliminary experiments, we have chosen 600 iterations in all cases, and parameters (SE_{min}, SE_{gap}) as $(40, 20)$, $(300, 100)$ and $(600, 300)$ for sizes 20, 50 and 150 respectively. Here is important to remark that SHRP runs faster than SVC because the stochastic algorithm that calculates patterns in SHRP is expected to be faster than the exact algorithm used in SVC. The value of parameter P is 0.5 for all instances. Table 6 summarizes the results of SRHP. In all three cases, it reaches a number of cuts between the best and worst results reached by SVC in 5 experiments, over 200 iterations each, with different values

Table 5 Summary of results of SVC over the problem classes 1, 4 and 9

class	1	4	9
z^{1D-CSP}	373.02	896.9	2669.56
[best, worst]	[373.66, 374.18]	[897.74, 898.92]	[2670.94, 2672.53]
$[\rho \text{ best}, \rho \text{ worst}]$	[1.04, 1.005]	[1.03, 1.005]	[1.02, 1.005]
time (sec.) 200 it.	7.2	39.6	456.0
$\rho \in [1.01, 1.03]$ chosen uniformly every 100 iterations, 2000 iterations all			
	373.48	899.24	2676.71
$\rho \in [1.01, 1.03]$ chosen uniformly every 5 iterations, 200 iterations all			
	373.52	899.31	2676.77

of parameter ρ . Moreover, SHRP outperforms SVC when this procedure iterates 2000 times, the parameter ρ varying uniformly every 100 iterations. As SHRP uses the same parameters for instances of equal size, in particular the same value of parameter P in all cases, we claim that SRHP is less dependent on parameter adjustment than SVC. Moreover, as we can observe in Table 6, even with a number of 50 iterations, SHRP is able to reach reasonably good solutions in all cases. Also, for SVC to reach better solutions than SHRP, it seems to be necessary to invest a considerable amount of time and effort in parameter tuning.

In Table 6, we also report results from similar experiments with SHRP by computing the set SE , of candidate patterns, by uniform selection of widths instead of selecting them in proportional ratio with the order's size. The results on number of cuts were a little bit better with the first method; even though for class 6, the results were better with the second one; but the number of patterns is clearly lower with the first method in all cases, with only one exception in class 7 where both methods produce similar results. Here it is important to remark that instances of class 7 have orders of small size as they take values uniformly in $S_1 = [1, 10] \cup \mathbb{Z}$.

Finally, we report results from similar experiments with parameter $P = 0$, i.e. without penalizing the number of rolls in the value of a pattern. In this case the results were really bad, as can be observed in the last rows of the table. Not only the number of cuts and patterns were much worse, but the required running times were much larger as well.

We have also experimented with the remaining set of instances. The results are summarized in Table 7. Here we report results from all instances of size 50 (instances 2 to 8) that correspond to classes $(50, v^1, S_2)$, $(50, v^2, S_2)$, $(50, v^3, S_2)$, $(50, v^4, S_2)$, $(50, v^5, S_2)$, $(50, v^3, S_1)$ and $(50, v^3, S_3)$ respectively. In particular, we show the best and worst results from SVC when it iterates 200 times, with fixed values of ρ and with this parameter varying uniformly every 5 iterations, and also the results from SHRP running with the above parameters. As in Table 7 we can observe that the number of cuts is also between the best and worst values reached by SVC with only one exception in class 5. Here it is worth to remark the variations of time consumed by procedures for different instances. The time required by SVC over class 2 is much larger than the time for the remaining ones. This is because instances of class 2 have orders with low widths in average, as they are taken uniformly in the interval $[0.01L, 0.2L) \cap \mathbb{Z}$. Thus, feasible

Table 6 Summary of results of SHRP over the problem classes 1, 4 and 9

class	1	4	9
z^{1D-CSP}	373.02	896.90	2669.56
600 iterations			
Number of cuts	373.32	898.63	2671.92
Number of patterns	18.30	46.16	134.03
time (sec.) 600 it.	3.75	21.02	320.00
3000 iterations			
Number of cuts	373.18	898.24	2671.42
Number of patterns	18.47	45.74	134.06
50 iterations			
Number of cuts	373.68	898.68	2673.24
Number of patterns	19.28	46.92	134.65
600 iterations, computing SE by uniform selection			
Number of cuts	373.29	898.66	2672.07
Number of patterns	18.97	48.17	141.62
time (sec.) 600 it.	3.49	20.00	323.10
600 iterations, $P = 0$			
Number of cuts	381.23	944.91	2855.65
Number of patterns	21.23	52.58	149.51
time (sec.) 600 it.	4.86	76.21	770.16

patterns have a larger number of items than patterns of the remaining instances, and so the exact procedure used in SVC to compute the next pattern has a larger number of possibilities to choose from. However, this fact is not an inconvenient for SHRP, as it takes a similar time than in other classes. In general, SHRP takes a time in direct ratio with the number of patterns.

6 Final Remarks

In this work, we have modelled a real problem, from a company that manufactures plastic rolls, as a variant of the one dimensional cutting stock problem (1D-CSP). The formulation extends the 1D-CSP by considering that different orders might have the same width, a number of additional constraints, and also four objective functions, in addition to the number of cuts, which is the main one. The new constraints establish that overproduction should be restricted to a predefined stock and also that in a pattern there are a minimum width and a maximum number of items. The new objective functions, according to the hierarchical order given by the experts, are the setup cost, the amount of stock generated, the completion times weighted by orders' priority and the number of open stacks. The setup cost is not computed simply as the number of different patterns. Instead, we consider the actual cost of changing cutting knives and pressure rollers in the real machine, as it is considered by the technicians. The amount of stock generated is taken as an objective function to be maximized because, after

Table 7 Summary of results of SVC and SHRP over the problem classes 2, 3, 4, 5, 6, 7, 8

class	2	3	4	5	6	7	8
z^{1D-CSP}	261.56	510.21	896.90	749.44	1173.18	90.10	1341.48
SVC							
Best No. of cuts	261.56	510.24	897.74	749.85	1173.65	90.18	1342.60
Worst No. of cuts	261.69	510.91	899.31	750.09	1179.29	90.29	1345.05
Time (sec.) 200 it.	134.40	55.20	39.60	60.00	39.60	30.00	40.80
SRHP (selection in direct ratio with b'_i)							
No. of cuts	261.59	510.59	898.63	751.89	1174.02	90.26	1344.26
No. of patterns	30.56	38.98	46.16	46.94	49.05	32.64	47.26
Time (sec.) 600 it.	12.32	11.72	21.02	14.35	20.72	16.03	21.58
SRHP (uniform selection)							
No. of cuts	261.58	510.58	898.66	751.85	1173.39	90.27	1344.80
No. of patterns	38.50	43.79	48.17	48.78	49.07	32.57	49.23
Time (sec.) 600 it.	16.15	13.52	20.00	15.65	19.16	16.15	18.22

minimizing the number of cuts, it is a good idea to make use of some of the trim-loss for generating stock rolls that will probably serve for future orders.

In order to solve the problem, we have developed a software application based on a GRASP algorithm. The generation phase of this algorithm solves a simplified version by means of a heuristic procedure termed SHRP, and in the repairing phase the simplified solution is transformed into a real solution. We have showed how the GRASP algorithm works on a small real instance. The application has been integrated in the information system of the company and is currently under exploitation. It has been registered as a trade mark of the company ERVISA with title “An intelligent system to compute cutting plans in manufacturing plastic rolls”. The application allows the expert to select a set of pending orders of the same product from the database and also to define a stock. Then, it computes a number of cutting plans. The selected cutting plan is saved and passed to the technicians to program the cutting machine. All orders are given before the cutting plan is obtained and dynamic planning was not considered. However, this possibility could be easily introduced as the time required for the machine to perform a cut is usually much longer than the time required for calculating a new plan. Typically, a cut takes a time of about 20 minutes and a cutting plan includes a number of cuts between 30 and 1000.

The key component of the software application is the procedure SHRP. This is similar in many aspects to other recent heuristic approaches, like SVC proposed by Belov and Scheithauer (2007) for the classic 1D-CSP, and the sequential heuristic proposed in Song et al (2006) for a variant termed 1.5-dimensional cutting stock problem. For the purpose of comparing SHRP with other approaches of the literature, we have applied it to the classic 1D-CSP and compare with SVC on the benchmark problems proposed in Belov and Scheithauer (2007). The results reported in section 5 show that SHRP produces results comparable to SVC and that SHRP is less dependent on parameter adjustment.

As future work, we plan to develop new strategies to transform simplified solutions into real solutions. To do so, we will consider the problem in the framework of multi-

objective optimization and will use some strategy, such as a genetic algorithm, to obtain the set of non-dominated solutions. In this way, the experts could choose the appropriate solution in function of their preferences at a given time. Also, we plan developing a branch and bound algorithm similar to the algorithm proposed in Belov and Scheithauer (2007) for the 1D-CSP. In this algorithm we will use SHRP to obtain upper bounds along the search, as Belov and Scheithauer used SVC in their branch-and-cut-and-price algorithm. We expect SHRP to be an efficient method to obtain upper bounds as it reaches good solutions even when running for a small number of iterations. In order to improve SHRP, we will try to design a systematic method to establish parameters from problem characteristics. In this work, we have given values for parameters such as P , SE_{min} and SE_{gap} , that worked well for a family of problems, but the values could be different for instances with different characteristics. Therefore, a systematic method that helps us to establish these values should be very interesting.

References

- Belov G, Scheithauer G (2003) The number of setups (different patterns) in one-dimensional stock cutting. Technical Report, Desden University
- Belov G, Scheithauer G (2006) A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *European Journal of Operational Research* 171:85–106
- Belov G, Scheithauer G (2007) Setup and open stacks minimization in one-dimensional stock cutting. *INFORMS Journal of Computing* 19(1):27–35
- Valério de Carvalho JM (2002) Lp models for bin packing and cutting stock problems. *European Journal of Operational Research* 141:253–273
- Chen CLS, Hart SM, Tham WM (1996) A simulated annealing heuristic for the one-dimensional cutting stock problem. *European Journal of Operational Research* 93:522–535
- Foerster H, Wscher G (1999) Pattern reduction in one-dimensional cutting stock problems. In: *Proceedings of the 15th Trienal Conference of the International Federation of Operational Research Societies*.
- Gilmore PC, Gomory RE (1961) A linear programming approach to the cutting stock problem. *Operations Research* 9:849–859
- Gilmore PC, Gomory RE (1963) A linear programming approach to the cutting stock problem. *Operations Research* 11:863–888
- Goulimis C (1990) Optimal solutions for the cutting stock problem. *European Journal of Operational Research* 44:197–208
- Gradisar M, Resinovic G, Kljajic M (2002) Evaluation of algorithms for one-dimensional cutting. *Computers and Operations Research* 29:1207–1220
- Haessler RW (1975) Controlling cutting patterns changes in one dimensional trim problems. *Operations Research* 23(3):483–493
- Haessler RW (1991) Cutting stock problems and solution procedures. *European Journal of Operational Research* 54:141–150
- Johnston RE (1986) Rounding algorithm for cutting stock problems. *Journal of Asian-Pacific Operations Research societies* 3:166–171
- Resende M, Ribeiro G (2002) *Greedy randomized adaptive search procedures.*, Kluwer Academic Publishers, pp 219–249

- Song X, Chu CB, Nie YY, Bennell JA (2006) An iterative sequential heuristic procedure to a real-life 1.5-dimensional cutting stock problem. *European Journal of Operational Research* 175:1870–1889
- Suliman SMA (2001) Pattern generating procedure for the cutting stock problem. *Int J Production Economics* 74:293–301
- Umetani S, Yagiura M, Ibaraki T (2003) One-dimensional cutting stock problem to minimize the number of different patterns. *European Journal of Operational Research* 146:388–402
- Vahrenkamp R (1996) Random search in the one-dimensional cutting stock problem. *European Journal of Operational Research* 95:191–200
- Vanderbeck F (2000) Exact algorithm for minimizing the number of setups in the one-dimensional cutting stock problem. *Operations Research* 48:915–926