

Solving Real-World Cutting Stock-Problems in the Paper Industry

– Mathematical Approaches, Experience and Challenges –

Julia Kallrath^{a,b}, Steffen Rebennack^{c,*}, Josef Kallrath^d, Rüdiger Kusche^e

^aGSEmbH, Oranienpassage 1, 57258 Freudenberg, Germany

^bh_da Hochschule Darmstadt, Schöfferstr. 3, 64295 Darmstadt, Germany

^cColorado School of Mines, Division of Economics and Business, Golden, CO 80401, USA

^dUniversity of Florida, Dept. of Astronomy, Gainesville, FL 32611, USA

^eGSEmbH, Oranienpassage 1, 57258 Freudenberg, Germany

Abstract

We discuss cutting stock problems (CSPs) from the perspective of the paper industry and the financial impact they make. Exact solution approaches and heuristics have been used for decades to support cutting stock decisions in that industry. We have developed polyolithic solution techniques integrated in our ERP system to solve a variety of cutting stock problems occurring in real world problems. Among them is the simultaneous minimization of the number of rolls and the number of patterns while not allowing any overproduction. For two cases, CSPs minimizing underproduction and CSPs with master rolls of different widths and availability, we have developed new column generation approaches. The methods are numerically tested using real world data instances. An assembly of current solved and unsolved standard and non-standard CSPs at the forefront of research are put in perspective.

Keywords: paper industry, cutting stock, roll production, format production, column generation, column enumeration, operations research, stochastic demand, real-world optimization

1. Introduction

The pulp and paper industry plays an important role worldwide. There are in the order of 3000 paper mills, which produced a total of 394 million tons of paper and paperboard, in 2010. Europe (including Russia) has approximately 900 paper mills, while Germany has about 180. The largest producer in the world is the Finnish UPM group with an annual tonnage of 12.7 million tonnes, followed by Stora Enso with 11.8 million tons and by International Paper with 9.7 million tonnes per year. Santos and Almada-Lobo (2012) report that in Portugal the pulp and paper industry contributes over

4% of the GDP and 5% of the active employees. As it is subject of both local and global environmental discussions, effective planning and cutting stock techniques lies at the very heart of the operational performance of its manufacturing organizations.

Exact solution approaches and heuristics have been used for decades to support cutting stock decisions in the paper industry. In the standard cutting stock problem (CSP), the problem input is given by a set of item sizes and demands, and by a set of master rolls of given widths; the simplest case consists of only one type of master rolls. The task is to decide on how many master rolls are cut to a certain pattern in order to minimize the total number of master rolls used.

The pattern minimization problem (PMP) is a strongly NP-hard cutting problem, which seeks a cutting plan with the minimum number of different patterns, *cf.* McDiarmid (1999). This objective, relevant when changing from one pattern to another, involves a cost for setting up the cutting

*Corresponding author

Email addresses: julia.kallrath@h-da.de (Julia Kallrath), srebenna@mines.edu (Steffen Rebennack), kallrath@astro.ufl.edu (Josef Kallrath), ruediger.kusche@gse-software.de (Rüdiger Kusche)
URL: <http://www.fbmn.h-da.de/~kallrath/> (Julia Kallrath), <http://www.rebennack.net> (Steffen Rebennack), <http://astro.ufl.edu/~kallrath> (Josef Kallrath)

machine, *i.e.*, adjusting the cutting knives. When the minimization of the number of different patterns is done by assuming that no more than the minimum number of rolls can be used, the problem is also referred to as the cutting stock problem with setup costs.

The international working group SICUP (*Special Interest Group on Cutting and Packing*) founded by Gerhard Wäscher in 1988, focuses on cutting stock and packing problems and is a platform for more than 200 practitioners and scientists to exchange ideas on these topics. In 2004, SICUP became the EURO working group ESICUP (*EURO Special Interest Group on Cutting and Packing*).

The main contributions of this paper can be classified into two categories:

Mathematical optimization: For 1D CSPs with two criteria, minimizing the number of rolls and the number of patterns, we develop an Exhaustion Method (Sect. 3.4), a column generation approach allowing underproduction (Sect. 3.5.4) and column generation approach incorporating master rolls with different widths and limited availability (Sect. 3.5.5). We present a novel polyhithic¹ solution method towards 2D trim-loss minimization (Sect. 4). Furthermore, we share real data in a 1D cutting stock benchmark data set (Sect. 3.6.1). For software products, it is not untypical to combine various basic algorithms to consistently provide solutions in acceptable time, with many empirical rules, or even rules of thumb, to decide which algorithms to use in each circumstance. We disclose this information instead of keeping it as a commercial secret, to provide evidence that there is more exact optimization and less heuristics involved as one might expect.

Managerial insights for the paper industry:

We present real-world aspects relevant to the paper industry, which have seen only little treatment in the scientific literature (Sect. 3.5). We assemble current cutting-edge standard and non-standard cutting stock problems relevant to the paper industry (Sect.

5) and illuminate at length the variants and issues present in real-world problems. We discuss the financial impact mathematical programming-based solutions to cutting stock problems have in the paper industry (Sect. 6).

The remainder of the paper is structured as follows: After a literature review in Section 2, we discuss the 1D CSP and its variants in Section 3 along with different solution techniques. A presentation of 2D polyhithic solution methods in Section 4 is followed by a discussion of current-edge CSPs in Section 5 and our views on optimization in the paper industry in Section 6. Conclusions are in Section 7. Two appendices, post-processing (Appendix A) and guidelines on how to derive the pricing problems (Appendix B) complete this paper.

2. Literature Review

There is a rich body of literature available on CSPs; *cf.* Haessler and Sweeney (1991) and Haessler (1992) for reviews on 1D cutting stock problems and solution procedures. We find heuristic solution approaches (*cf.* Haessler (1971)), exact MILP-models (*cf.* Johnston and Sadinlija (2004)), column generation approaches, among them the classical paper by Gilmore and Gomory (1961), Branch&Price algorithms (*cf.* Belov and Scheithauer (2006)), reviews as by Amor (2005) who put column generation and Branch&Price algorithms in perspective, and classification papers (*cf.* Dyckhoff (1990) and Wäscher et al. (2007)).

Most of the approaches described in the literature for solving the PMP are based on heuristics. As the PMP has been proven strongly NP-hard by McDiarmid (1999), it is not a surprise that solving the problem exactly has been a real challenge, and only very few exact solution methods have been reported so far in the literature; among them Vanderbeck (2000). Alves et al. (2009) explore an integer programming model that can be solved using column generation, and they describe different strategies to strengthen it, among which are constraint programming and new families of valid inequalities. Lower bounds for the pattern minimization problem are derived from the new integer programming model, and also from a constraint programming model.

Beyond a vast body of literature on the standard CSP, there are a few publications on a generalized CSP with great practical significance: The multiple-width CSP with master rolls of different

¹The term *polyhithic* has been coined by Kallrath (2009a) and explained in greater detail in Kallrath (2011); it refers to modeling and solution approaches in which mixed integer or nonconvex nonlinear optimization problems are solved by tailor-made methods involving several models and/or solve statements or algorithmic components.

widths (and equal lengths assumed to be infinite). An early work on this topic is by Holthaus (2002), who solves the relaxation of the CSP by the column generation technique and uses three procedures for rounding the solution, leading in a final residual problem, which is solved by an ILP-solver. Although his technique is suitable for solving medium-size and large instances of the one-dimensional CSP, the paper does not consider supply limitation on the different stock lengths availability. Alves and de Carvalho (2007) developed strategies to stabilize and accelerate the column generation method by introducing dual-optimal inequalities, reducing the number of column generation iterations and run time. Finally, Poldi and Arenales (2009) provide a heuristic to solve the CSP with multiple stock lengths with limited availability.

Although production planning or scheduling and CSPs are usually treated separately, we find early articles in which both aspects are combined; *cf.* Haessler and Talbot (1983) or Li (1996) who provide LP-based and non-LP-based heuristics to solve 2D multi-job cutting stock problems with due dates and release dates. The combined cutting stock and lot-sizing problem in industrial processes has attracted several authors in the last decade, among them Arbib and Marinelli (2005), Gramani and França (2006), Yanasse and Pinto Lamosa (2007), Trkman and Gradisar (2007), Poltroniere et al. (2008), Gramani et al. (2009) and most recently Reinertsen and Vossen (2010) who treat the 1D CSP with due dates. Trkman et al. (2009) treat cutting stock as a continuous business process which is incorporated into an entire supply chain.

General cutting and packing problems are related to CSPs. The most important difference between cutting and packing problems is that in cutting problems, the number of objects are given and the task is to minimize trim-loss or area, while packing problems aim to fit as many objects as possible in a predefined area or volume. For example, one may want to cut orientation free polygons (Kallrath, 2009b) or ellipses (Kallrath and Rebennack, 2014) into one rectangle, or circles into several rectangles (Rebennack et al., 2009). A significant difference between these cutting problems cited and the 2D cutting problems described in Section 4 is that the latter allow only a horizontal or vertical orientation of the objects to be cut.

We conclude our literature review by pointing the reader to a few articles which give some excellent insights into the field: Rodríguez and Vecchi-

etti (2008) for practical application with very good illustrations, and similarities to our 2D problem described in Section 4, Harjunkoski et al. (1998) and Pörn et al. (1999) for exact MILP and MINLP approaches, and also, a very recent paper on heuristics by Cui and Zhao (2013).

3. 1D Cutting Stock Problem

Our discussion of the one-dimensional cutting stock problem starts with the standard problem formulation in Section 3.1, followed by three solution methods: the widely used approach by Gilmore & Gomory (Sect. 3.2), a column enumeration (Sect. 3.3), and an Exhaustion Method (Sect. 3.4). We summarize important practical aspects for one-dimensional CSPs for the paper industry and present extensions to the column generation approach addressing these practical aspects (Sect. 3.5). We conclude this section with some computational benchmarking (Sect. 3.6).

3.1. The Standard Problem and its Mathematics

The mathematical model for minimizing the number of rolls or trim-loss in the standard problem with one master roll of width B is characterized by the following indices, data and variables.

3.1.1. Indices

$p \in \mathcal{P} := \{p_1, \dots, p_{N^P}\}$ cutting patterns; N^P is the number of patterns in \mathcal{P} . If N^P is not known, as it happens in the Gilmore and Gomory approach, we set N^P to a sufficiently large number.

$i \in \mathcal{I} := \{i_1, \dots, i_{N^I}\}$ given orders or widths; N^I is the number of (orders) widths in \mathcal{I} .

3.1.2. Input Data

B [L] width of the master rolls (raw material rolls).

D_i [-] demand; the requested number of pieces of width i .

W_i [L] width of order type i .

3.1.3. Integer Variables

$\mu_p \in \mathbb{N}_0 := \{0, 1, 2, 3, \dots\}$ [-] indicates how often pattern p is used; $\mu_p = 0$ if p is not used.

$\alpha_{ip} \in \mathbb{N}_0$ [-] indicates how often order type i is contained in pattern p ; $0 \leq \alpha_{ip} \leq D_i$.

3.1.4. Mathematical Programming Model Formulation

A suitable objective function

$$z^* := \min_{\alpha_{ip}, \mu_p} \sum_{i \in \mathcal{I}} \sum_{p \in \mathcal{P}} f(\alpha_{ip}, \mu_p) \quad , \quad (1)$$

is subject to the restrictions (fulfillment of the demand)

$$\sum_{p \in \mathcal{P}} \alpha_{ip} \mu_p = D_i \quad , \quad \forall i \quad , \quad (2)$$

feasibility of pattern p

$$\sum_{i \in \mathcal{I}} W_i \alpha_{ip} \leq B \quad , \quad \forall p \quad , \quad (3)$$

and the integrality constraints

$$\alpha_{ip} \in \mathbb{N}_0 \quad , \quad \forall \{ip\} \quad (4)$$

and

$$\mu_p \in \mathbb{N}_0 \quad , \quad \forall p \quad . \quad (5)$$

Formulated via (1)-(5), the standard CSP is a mixed integer nonlinear (nonconvex) optimization problem (MINLP), a problem class which is difficult in itself. As the problem may easily encounter several million variables α_{ip} , it cannot be solved efficiently in this form. Another complication results from equation (2) with exact demand fulfillment, as it is rare to find feasible solutions to the CSP. Therefore, in practical situations, (2) is relaxed to

$$D_i^L \leq \sum_{p \in \mathcal{P}} \alpha_{ip} \mu_p \leq D_i^U \quad , \quad \forall i \quad , \quad (6)$$

with lower and upper bounds, D_i^L and D_i^U , on the demand D_i . Usually, underproduction is less accepted as light overproduction; see also the discussion in Section 3.5.4.

3.2. Gilmore & Gomory Approach (GGA)

The idea of column generation by Gilmore and Gomory (1961) is to dynamically add variables (“columns”) which are good candidates to be included in an optimal solution. This is achieved by decomposing the CSP into a master problem (MP) and a sub-problem (SP), also called *pricing problem*. For a predefined set of patterns $\mathcal{P}' \subseteq \mathcal{P}$, the MP decides how often each pattern has to be used and provides input data for the SP via dual information. It minimizes the number of rolls

$$\min_{\mu_p} \sum_{p \in \mathcal{P}'} \mu_p \quad , \quad (7)$$

with the demand-fulfill inequalities (note that it is allowed to produce more than requested)

$$\sum_{p \in \mathcal{P}'} N_{ip} \mu_p \geq D_i \quad , \quad \forall i \quad , \quad (8)$$

where N_{ip} is the number of times order i is contained in pattern $p \in \mathcal{P}'$. The integrality constraints

$$\mu_p \in \mathbb{N}_0 \quad , \quad \forall p \in \mathcal{P}' \quad . \quad (9)$$

complete the model. Replacing the integrality requirement (9) on μ_p by a non-negative constraint, we obtain the so-called *relaxed master problem* (RMP).

In the SP, new patterns (variables α_i , the multiplicity of width i) are calculated by exploiting the dual values π_i (pricing information) of the RMP associated with (8). The objective function

$$\min_{\alpha_i} \left(1 - \sum_{i \in \mathcal{I}} \pi_i \alpha_i \right) \quad ,$$

involves the integer variables α_i . To ensure that, in the new pattern, the roll width, B , and the number, K , of knives are not exceeded we add the knapsack inequalities

$$\sum_{i \in \mathcal{I}} W_i \alpha_i \leq B \quad , \quad \sum_{i \in \mathcal{I}} \alpha_i \leq K \quad . \quad (10)$$

Numerical experiments without the knife constraint are indicated by $K = \infty$. For completeness, we note the integrality conditions

$$\alpha_i \in \mathbb{N}_0 \quad , \quad \forall i \quad . \quad (11)$$

The α_i become the N_{ip} coefficients of the new pattern in (8). In some cases, α_i could be additionally bounded, for instance, by the number, K , of available knives, or by the demand, D_i .

Once the optimal objective function value of the SP is non-negative, then the RMP for pattern set \mathcal{P}' has been solved to optimality over *all* possible patterns. The optimal objective function value of the RMP provides a lower bound on z^* ; solving the MP for the available set of patterns \mathcal{P}' yields an upper bound on z^* .

In this context, the absolute difference (*i.e.*, gap) between the RMP and the MP, once GGA converges, is of great interest. A well known conjecture states that the 1D cutting stock problem (when minimizing the number of patterns used) has the so-called modified round-up property, *i.e.*, the gap

is at most 2, *cf.* Scheithauer and Terno (1995). No instance of the 1D cutting stock problem has been reported so far which has a greater gap. To the best knowledge of the authors, the conjecture has not been proven, yet.

3.3. Column Enumeration

As the expression “column enumeration” (CE) suggests, the set of possible columns (*e.g.*, patterns) is enumerated. As such, CE is a special variant of column generation and is applicable when a small number of columns is sufficient. For instance, this is the case in real-world cutting stock problems when it is known that the optimal solutions have only a small amount of trim-loss, eliminating most of the patterns. CE naturally leads to a type of selecting columns or partitioning models: problem (7)-(9) with \mathcal{P}' being the set of all generated columns. Despite the limitations with respect to the number of columns, CE has some advantages: No pricing problem, easily applicable to MILP problems, much easier to implement (compared to column generation), and allows the straight forward incorporation of demand stochasticities (*cf.* Sect. 5.1). In the context of cutting stock problems, we sometimes can use the maximum permissible trim-loss to restrict the number of patterns to be considered in CE (*cf.* Sect. 3.5).

3.4. An Exhaustion Method

This method combines a constructive heuristic with exact MILP techniques. We illustrate the exhausting method by the CSP described in Section 3.1; assigning orders in a scheduling problem would be another example of an exhaustion approach. The elegant GGA is known for producing minimal trim-loss solutions with *many* patterns. Often this corresponds to setup changes on the machine and therefore is not desirable. A solution with a minimal number of patterns minimizes the machine setup costs of the cutter. Minimizing simultaneously trim-loss and the number of patterns is possible for small cases of a few orders only, exploiting the MILP model by Johnston and Sadinlija (2004). It contains two conflicting objective functions. Therefore one could resort to goal programming. Alternatively, we produce a pool of several parameterized solutions leading to different number of rolls to be used and patterns to be cut from. It is up to the user to choose the best solution from that pool; *cf.* Section 3.6.4.

Note that the Branch&Price algorithm described in Vanderbeck (2000) or Belov and Scheithauer (2006) can be used to solve the 1D CSP with minimal numbers of patterns. However, these methods are not easy to implement. Therefore, depending on the number of orders, we use the following approaches:

- V1: Direct usage of the model by Johnston and Sadinlija (2004), for a small number of orders, *e.g.*, $N^I \leq 14$ and $D_{\max} \leq 10$, to minimize the number of patterns or the number of rolls. In a preprocessing step, we compute valid inequalities as well as tight lower and upper bounds on the variables.
- V2: Exhaustion procedure in which we cover the orders and their demands by generating successively new patterns with maximal multiplicities (phase V2-1). If only a small number of orders is left over, we use V1 to minimize the number of rolls (phase V2-2). In phase V2-3, we exploit the MIPSTART feature of CPLEX and start with the best solution found (smallest number of patterns) and use again V1 to minimize the number of patterns with the aim to compute an improved lower bound, N_{2-}^P , smaller than our initial lower bound, N_{1-}^P , which we obtained by solving the corresponding bin packing problem (BPP, resulting from setting all CSP demands to 1), and to find a better solution.

3.4.1. Indices and Sets

In this model, we use the indices listed in Johnston and Sadinlija (2004):

$i \in \mathcal{I} := \{i_1, \dots, i_{N^I}\}$ the index set of (order) widths.

$p \in \mathcal{P} := \{p_1, \dots, p_{N^P}\}$ the set of all possible patterns; $N^P \leq N^I$. The patterns are generated by V1, or dynamically by maximizing the multiplicities of a pattern used. Note that in Johnston and Sadinlija (2004), the index j is used instead of p .

$k \in \mathcal{K} := \{k_1, \dots, k_{N^K}\}$ the multiplicity index to indicate how often a width is used in a pattern. The multiplicity index can be bounded by the ratio of the widths of the orders and given rolls.

3.4.2. Variables

The following integer or binary variables are used:

$\alpha_{ip} \in \mathbb{N}$ [-] specifies how often width i occurs in pattern p ; $0 \leq \alpha_{ip} \leq D_i$ (item-in-pattern multiplicity).

$\alpha_{ipk} \in \mathbb{N}$ [-] auxiliary variable connected to δ_{ipk} ; $0 \leq \alpha_{ipk} \leq D_{\max} := \max_i D_i$.

$\delta_p^A \in \{0, 1\}$ [-] indicates whether pattern p is used at all.

$\delta_{ipk} \in \{0, 1\}$ [-] indicates whether width i appears in pattern p at level k ; $\delta_{ipk} = 0$ implies $\alpha_{ipk} = 0$.

$\mu_p \in \mathbb{N}$ [-] specifies how often pattern p is used. If pattern p is not used, we have $\mu_p = \delta_p^A = 0$.

3.4.3. The Idea of the Exhaustion Method

In our exhaustion procedure, we cover the orders and their demands by generating successively new patterns with maximal multiplicities. If only a small number of orders is left over, we switch to V1. This method is parameterized by the initial permissible percentage waste W_{\max} , $1 \leq W_{\max} \leq 99$. To populate the pool, we use six parameterizations with the following values: $W_{\max} = 20, 15, 10, 8, 6$ and 4 . The usage of different parametrizations is motivated by the hope to obtain different solutions. This allows the user to select the best combination of number of rolls and number of patterns.

We start with iteration $m = 0$. In each iteration m , we generate at most 2 new patterns by maximizing the multiplicities of these patterns allowing no more than a maximum percentage waste, W_{\max} , relative to the width of the master roll. The solution generated in iteration m is preserved in iteration $m + 1$ by fixing the appropriate variables. If the problem turns out to be infeasible (this may happen if W_{\max} turns out to be restrictive; we observed this to happen occasionally for $W_{\max} < 8$), we increase the permissible waste by 10% and proceed to the next iteration. If only a few orders remain, we switch to V1 to cover the remaining unsatisfied orders; note that V1 works without the W_{\max} restriction.

Our model is based on the MILP model developed by Johnston and Sadinlija for solving the 1D-CSP problem, especially on their inequalities (1,2,3,5-8). Their main idea is to replace the nonlinear terms in (2) by binary variables δ_{ipk} , which take

value 1 if item i occurs in pattern p with multiplicity k . Similar transformations have been presented before by Harjunkoski et al. (1998). The model by Johnston and Sadinlija can work efficiently, if we expect to have only reasonably small item-in-pattern multiplicities k and not too many items and patterns.

It helps, and is also necessary, to provide lower and upper bounds, N_-^P and N_+^P , on the number of patterns expected to be used in an optimal solution. If N^I is the number of items (order widths), then $N_{1+}^P := N^I$ is a weak upper bound. We compute a weak lower bound, N_{1-}^P , by applying the GGA towards a BPP associated with the CSP problem.

The great strength of the model by Johnston and Sadinlija is that it allow us to implement different objective functions and constraints much easier than in a MINLP model or in column generation approaches. Therefore, it even allows for sequencing production by exploiting a one-to-one correspondence between pattern and manufacturing sequence. We briefly summarize the relevant relations we implemented in our exhaustion; note that we partially adjusted the nomenclature used by Johnston and Sadinlija to be consistent with our paper.

Let binary variable δ_p^A indicating whether or not pattern $p \in \mathcal{P}$ is used in the optimal solution. The pattern multiplicity, μ_p , is subject to lower and upper bounds, M^L and M^U , *i.e.*,

$$\mu_p \geq M^L \quad , \quad p \in \mathcal{P}_{\text{act}} := \{1, \dots, N_-^P\} \quad (12)$$

and ($p \in \mathcal{P}_{\text{pot}} := \{N_-^P + 1, \dots, N_+^P\}$)

$$M^L \delta_p^A \leq \mu_p \leq M^U \delta_p^A. \quad (13)$$

Johnston and Sadinlija leave it to the user to set M^L and M^U . Setting $M^L = 1$ is the easiest choice; we use $M^U = \max_i \{D_i\}$.

The binary variables δ_{ipk} are accompanied by integer variables α_{ipk} which are constructed in such a way that $\delta_{ipk} = 1$ implies $\alpha_{ipk} = \mu_p$. Instead of a fixed demand D_i , Johnston and Sadinlija allow for bounded under- and overproduction, *i.e.*, $D_i^L \leq D_i \leq D_i^U$, and thus

$$D_i^L \leq \sum_p \sum_k k \alpha_{ipk} \leq D_i^U \quad , \quad \forall i \quad . \quad (14)$$

The various binary and integer variables δ_{ipk} , α_{ipk} ,

and μ_p are connected by

$$\alpha_{ipk} \leq M^U \delta_{ipk} \quad , \quad \forall \{ipk\} \quad , \quad (15)$$

$$\sum_k \delta_{ipk} \leq 1 \quad , \quad \forall \{ip\} \quad , \quad (16)$$

$$\sum_k \alpha_{ipk} \leq \mu_p \quad , \quad \forall \{ip\} \quad , \quad (17)$$

and

$$M^U \sum_k \delta_{ipk} - \sum_k \alpha_{ipk} + \mu_p \leq M^U \quad , \quad \forall \{ip\} \quad . \quad (18)$$

Inequalities (15) and (16) guarantee that only one δ_{ipk} and its corresponding α_{ipk} can be selected for each pair (ip). Inequalities (17) and (18) guarantee that $\alpha_{ipk} > 0$ implies $\alpha_{ipk} = \mu_p$.

The proper design of the patterns is ruled by the knapsack inequalities

$$\sum_i \sum_k W_i k \delta_{ipk} \leq B \quad , \quad \forall p \quad , \quad (19)$$

and

$$\sum_i \sum_k W_i k \delta_{ipk} \geq \frac{100 - W_{\max}}{100} B \quad , \quad \forall p \quad . \quad (20)$$

We add additional constraints to the model, *e.g.*, the symmetry breaking inequality

$$\delta_p^A \geq \delta_{p+1}^A \quad , \quad \forall p \quad , \quad (21)$$

which ensures that pattern $p + 1$ can only be used if p is used. The symmetry breaking inequality

$$\mu_p \geq \mu_{p+1} \quad , \quad \forall p \quad , \quad (22)$$

orders the patterns according to their multiplicities.

We exploit the Johnston and Sadinlija model by three objective functions. In V2-1, we maximize the multiplicities of the patterns generated

$$\max_{\mu_p} \sum_{p \in \mathcal{P}_{\text{pot}}} \mu_p \quad ,$$

while in (14) we set $D_i^L = 0$ and $D_i^U = D_i'$, where D_i' is the number of remaining orders of width index i . In V1, we minimize the number of patterns

$$N^P := \min_{\delta_p} \sum_{p \in \mathcal{P}'} \delta_p = P^L + \min_{\delta_p^A} \sum_{p \in \mathcal{P}_{\text{pot}}} \delta_p^A \quad ,$$

subject to $D_i^L = D_i^U = D_i$, and in V2-2, we minimize the number of rolls

$$\min_{\mu_p} \sum_{p \in \mathcal{P}_{\text{pot}}} \mu_p \quad (23)$$

subject to $D_i^L = D_i^U = D_i$. The model is completed by the integrality conditions ($\forall \{ik\}$ and $p \in \mathcal{P}_{\text{pot}}$)

$$\mu_p, \alpha_{ipk} \in \mathbb{N}_0 \quad , \quad \delta_p^A, \delta_{ipk} \in \{0, 1\} \quad .$$

The model is applied several times with $\alpha_{ipk} \leq D_i'$. Especially, the model has to fulfill the relationships ($\forall \{ik\}$ and $p \in \mathcal{P}'$)

$$k\alpha_{ipk} > D_i' \implies \alpha_{ipk} = 0 \quad \wedge \quad \delta_{ipk} = 0$$

as well as ($\forall \{ik\}$ and $p \in \mathcal{P}_{\text{pot}}$)

$$\alpha_{ipk} \leq \left\lceil \frac{D_i'}{k} \right\rceil \quad \text{and} \quad \alpha_{ipk} \leq \left\lceil \frac{D_i' + S_i}{k} \right\rceil \quad ,$$

where S_i denotes the permissible overproduction.

This Exhaustion Method provides an improved upper bound, $N_{2+}^P = N_{\min}^P$, $N_{2+}^P \leq N_{1+}^P$, on the number of patterns, where N_{\min}^P is the number of patterns in the best solution (smallest number of patterns) of the pool.

3.4.4. Phase V2-3: Computing the Lower Bound on the Number of Patterns

To compute a lower bound on N_-^P , we apply two methods. The first method is to solve a BPP which is equivalent to minimizing the number of rolls in the original cutting stock problem described in Section 3.1 for equal demands $D_i = 1$. If solved with the column generation approach, this method is fast and cheap (for the cases we are interested in with up to 80 orders), but the lower bound, N_{1-}^P , obtained is often weak for the PMP, *cf.* Vanderbeck (2000)). The second method, used in phase V2-3, is to exploit the N_{\min}^P solution to use the exact model V1 for minimizing the number of patterns. This enables us to work with a smaller set of potential patterns $\mathcal{P}_{\text{pot}} = \{1, \dots, N_{\min}^P\}$. It is impressive to see how quickly the commercial solvers CPLEX and XpressMP improve upon the lower bound yielding N_{2-}^P , when we utilize the MIPSTART feature. For most examples with up to 50 orders we obtain $N_{2+}^P - N_{2-}^P \leq 2$, but in many cases $N_{2+}^P - N_{2-}^P = 1$ or even $N_{2+}^P = N_{2-}^P$. Sometimes, in step V2-3, we even find a better solution, *i.e.*, a solution with fewer patterns than N_{\min}^P .

3.5. Practical Aspects in the Paper Industry: Towards an Implementation at GSEmbH

Production in the paper industry is closely connected to the cutting machines and their properties, *e.g.*, the number of available knives, or the minimal

distance to the edges of the master rolls. The production philosophy in each company differs. Usually, the production planning problem and the cutting stock problem are not treated in one consistent MILP or MINLP model. As these two problems are inseparable, they must be solved somehow hand-in-hand, if not in one algorithm then at least iteratively. In some production planning heuristics, groups of orders are constructed, which become the input of a CSP. However, it is not unusual to find people working out the cutting stock pattern to leave out a few orders as they do not ideally fit into the patterns constructed or lead to patterns with too much trim-loss. These missing orders are considered in later production runs. In the strict sense, this means that underproduction is allowed. While the GGA in its standard form assumes that overproduction is allowed, in many practical instances, the planners are not amused about overproduction and do not really know what to do with the additional items. Efficient patterns with only a few percent trim-loss are an important issue which can easily conflict with other objectives, *e.g.*, to minimize the number of patterns used. All these practical issues require us to apply modifications to the GGA or to resort to other techniques.

3.5.1. Implementing the Gilmore & Gomory Column Generation Approach

The GGA requires a few modifications to avoid the production of unnecessarily many orders:

- Generation of the initial patterns (by solving a separate knapsack problem which maximizes the number of used widths in the pattern while ensuring that each width is used at most twice *i.e.*, $\alpha_i \leq 2, \forall i$). If it is desirable to work primarily with efficient patterns, then it is better not to restrict α_i artificially but only use the demand, D_i , for item i , in the inequality $\alpha_i \leq D_i, \forall i$, in order to fill the patterns to the maximum. The initial knapsack problem is solved N^I times. To guarantee feasibility of the initial master problem, we enforce for the i -th problem that width W_i is contained at least once in the pattern.
- When dynamically generating new patterns, we ensure that each width is not contained more often than it is ordered. However, this restriction cannot be used when only efficient patterns with the minimum waste are considered.

- Post-processing: Elimination of surplus items generated by the GGA as described in Appendix A.

Additional constraints lead to an extension of the GGA approach, which we refer generally to as *column generation approach* (CGA). These constraints may enter the master problems and/or the pricing problems. Constraints which influence the pattern design show up in the pricing problem. The restriction of the number of knives is one example. Another example is the requirement that very small strips need to be embedded by normal order widths. As long as such constraints do not require substantially more effort to solve the pricing problems they are harmless, *i.e.*, the additional constraints do not destroy the structure of the pricing problems. Constraints showing up in the master problem tend to produce more difficulties. An example are constraints counting and restricting the number of patterns used in the optimal minimal trim-loss solution. We discuss two additional examples and their mathematics in Sections 3.5.4 and 3.5.5.

In general, we observe that the pattern space generated by the GGA is not complete. Thus, using the generated patterns by the GGA approach and changing the objective function or imposing additional constraints may lead to sub-optimal solutions or an infeasible problem.

If the CGA, due to additional constraints or a modified objective function, does not work efficiently anymore, we resort to complete or partial column enumeration in which we construct reasonable patterns explicitly. Afterwards, a set partitioning problem is solved for a different objective function. However, this approach works only if not too many patterns are generated (in the order of 10^4 to $5 \cdot 10^4$); otherwise, it may become too difficult to solve the partitioning problem.

3.5.2. Efficient Patterns

The paper producers very often want to accept only *efficient patterns* which do not have more than a certain trim-loss, W_{\max} , derived from the tolerable percent trim-loss, $W_{\max}^{\%}$. Depending on the order situation, this constraint may lead to infeasible situations. This is illustrated by the following example (all data carry the same units): A master roll with width of $W = 266$, and order widths $W_1 = 140$, $W_2 = 138$, and $W_3 = 136$ lead to patterns with only one order width. This

in turn produces an unacceptable large strip-loss of $(266 - 140)/266 = 47.36\%$. Thus, for $W_{\max}^{\%} \leq 47.36\%$, the problem becomes infeasible.

In such cases, it is helpful to derive the smallest value, $W_{\min}^{\%}$, of $W_{\max}^{\%}$ for which a feasible solution exists at all. What happens in real life is that the planner changes the order spectrum to leave out one or several critical orders. Again, this selection can be supported by appropriate auxiliary models. We allow underproduction with respect to the given orders but our objective function is to minimize this underproduction. In order to achieve this, we generate the patterns by the GGA, followed by a master problem in which we minimize the underproduction. A conceptual problem with this approach is that we cannot be sure that we really produced all relevant patterns. Thus, we only obtain an upper bound on underproduction.

3.5.3. Exact Demand Satisfaction

The demand is usually not met exactly with the GGA, as the column generation can lead to overproduction. In cases, in which we apply the GGA, we eliminate superfluous orders where possible with the heuristics described in Appendix A. However, these simple heuristics cannot always eliminate overproduction. In these cases, we resort to the MILP model of Johnston and Sadinlija to compute exact demand solutions.

The simple inequality ($\forall i$ and $p \in \mathcal{P}'$)

$$N_{ip} \leq D_i$$

in the pricing problem can avoid some overproduction problems. If the initial patterns also obey this inequality, at least, it is impossible that the usage of a single pattern p with $\mu_p = 1$ exceeds demand.

Partitioning models adhere to the following upper bound on μ_p

$$\mu_p \leq \max_{i \mid N_{ip} > 0} \left\{ \left\lfloor \frac{D_i}{N_{ip}} \right\rfloor \right\}, \quad \forall p \in \mathcal{P}' .$$

This is, however, rather a numerical improvement when solving the master problem and does not strictly avoid overproduction. Adding the following upper bound on μ_p

$$\mu_p \leq \min_{i \mid N_{ip} > 0} \left\{ \left\lfloor \frac{D_i}{N_{ip}} \right\rfloor \right\}, \quad \forall p \in \mathcal{P}'$$

to the partitioning model may help, but can easily lead to infeasibility.

One might feel tempted to avoid overproduction by solving the CGA master problem with the exact demand constraint

$$\sum_{p \in \mathcal{P}'} N_{pi} \mu_p = D_i, \quad \forall i,$$

replacing (8). Unfortunately, this may also lead to an infeasible partitioning problem as the pattern space does not allow for this. The resulting GGA still converges theoretically. However, besides feasibility issues when solving the master problem, the gap might no longer be small.

The fundamental problem with exact demand fulfillment is to generate the correct pattern space. A Branch&Price procedure is able to achieve this but at significant effort, computationally as well as implementation wise. For completeness, we note that if overproduction is allowed, then ($p \in \mathcal{P}'$)

$$B - \sum_{i \in \mathcal{I}} N_{ip} \leq \min_i D_i .$$

3.5.4. Allowing Underproduction

In real-world CSPs, we experience at least two reasons for allowing or dealing with underproduction. The first reason is, that the group of orders could lead to patterns which are not easily accepted due to large trim-loss. People prefer to leave out some complicating orders or to fulfill them only partially. The second reason may occur in situations where overproduction is strictly forbidden. This may lead to additional patterns or patterns with large trim-loss. Again, in such situations it may be better to underfill demand.

However, conceptually, it is not trivial to model underproduction because it leads to conflicting objective functions. If we allow one pattern for each item and pattern multiplicity is not restricted, we can avoid underproduction completely. This example illustrates that we somehow need to balance underproduction versus the number of rolls we are willing to use, or equivalently, we need to balance underproduction versus trim-loss. Therefore, the objective function (24) contains a term which maximizes the production or cutting of items and minimizes the number of rolls weighted by their individual waste. The weighting factor ρ balances both aspects.

The objective of the master problem is now to maximize a weighted total production function

$$\max \sum_{p \in \mathcal{P}'} \left(\sum_{i \in \mathcal{I}} W_i N_{ip} - \rho W_p \right) \mu_p, \quad (24)$$

with factor ρ and pattern waste W_p , subject to the demand inequalities allowing underproduction

$$\sum_{p \in \mathcal{P}'} N_{ip} \mu_p \leq D_i \quad , \quad \forall i \quad , \quad (25)$$

as well as the integrality constraints (9).

The sub-problem remains structurally a knapsack-constrained MILP with objective function

$$\max_{\alpha} \sum_{i \in \mathcal{I}} (W_i - \pi_i) \alpha_i - \rho \left(B - \sum_{i \in \mathcal{I}} W_i \alpha_i \right) \quad ,$$

where π_i are the dual values of the demand inequality (25), subject to constraints (10) & (11). The term

$$B - \sum_{i \in \mathcal{I}} W_i \alpha_i^*$$

defines the waste W_p of the newly generated pattern p , corresponding to optimal solution α_i^* .

3.5.5. Master Rolls with Different Widths and Limited Availability

Instead of one roll type with width B and infinite length, we now consider N^R types of rolls with width B_r , $r \in \mathcal{R} = \{r_1, \dots, r_{N^R}\}$. Further, we assume that there are $N_r \in \mathbb{N}_0$ rolls of type r available on stock.

We modify the GGA as follows: The master problem contains the information of the limited availability of the rolls while the subproblems generate new patterns for each roll type. The subproblems separate with roll type r . This approach is similar to the work by Holthaus (2002). However, we consider a more involved objective function (leading to different pricing problems) and we limit the number of available master rolls.

If only one type of master rolls is available and if overproduction counts as waste, minimizing waste or number of rolls is equivalent. However, this is not true any longer for master rolls of different widths. When minimizing the number of rolls, the optimal solution is to use the master roll of largest width as often as possible. Therefore, we select as the objective function for the master problem the waste per pattern multiplied by the pattern multiplicity

plus the overproduction:

$$\begin{aligned} \min \sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}'_r} \left(B_r - \sum_{i \in \mathcal{I}} W_i N_{ip} \right) \mu_p \\ + \sum_{i \in \mathcal{I}} \left(\sum_{p \in \mathcal{P}'} N_{ip} \mu_p - D_i \right) \end{aligned} \quad (26)$$

$$\begin{aligned} = \sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}'_r} \left(B_r - \sum_{i \in \mathcal{I}} (W_i - 1) N_{ip} \right) \mu_p \\ - \sum_{i \in \mathcal{I}} D_i \quad . \end{aligned} \quad (27)$$

The master problem (7)-(9) gets extended by the following constraints

$$\sum_{p \in \mathcal{P}'_r} \mu_p \leq N_r \quad , \quad \forall r \quad , \quad (28)$$

where p is member of subset $\mathcal{P}'_r \subset \mathcal{P}'$, if pattern p is generated for roll type r ; *e.g.*, $\cup_{r \in \mathcal{R}} \mathcal{P}'_r = \mathcal{P}'$.

We obtain one knapsack-type subproblem for each roll type r with objective function

$$\begin{aligned} z_r^* := \min \left(B_r - \sum_{i \in \mathcal{I}} (W_i - 1) \alpha_i - \pi_r - \sum_{i \in \mathcal{I}} \pi_i \alpha_i \right) \\ = \min \left(B_r - \pi_r - \sum_{i \in \mathcal{I}} (W_i - 1 + \pi_i) \alpha_i \right) \quad , \end{aligned}$$

with dual variable π_r associated with constraints (28). The slightly modified feasibility constraint for a new pattern for each master roll type r reads

$$\sum_{i \in \mathcal{I}} W_i \alpha_i \leq B_r \quad , \quad \alpha_i \in \mathbb{N}_0 \quad , \quad \forall \{i\} \quad .$$

The column generation procedure converges, if

$$\forall r : \quad z_r^* \geq 0 \quad ,$$

otherwise, for each $z_r^* < 0$, one new pattern p for roll type r is generated (this yields N_{ip} and pattern p is included in \mathcal{P}'_r). Note that the computational effort is only linearly higher than in the column generation approach for the standard CSP. Instead of one pricing problem, we have to solve N_r of them. However, with the objective function change, we lose the nice modified rounding-up property. Therefore, we need to watch the gap between the objective function values of the last relaxed master problem and the final MILP master problem.

3.5.6. Alternative Objective Functions

Two examples of alternative objective functions are: (1) to minimize the number of patterns used, and (2) to minimize the number of rolls subject to the constraints that at most a certain number of patterns could be used.

A change of objective function (or problem structure) may lead to a large gap between the relaxed master problem and the master problem, because the pattern space generated by a CGA is not complete. In this case, one can implement a Branch&Price algorithm or resort to the model by Johnston and Sadinlija (2004).

3.5.7. Stability and Robustness of Software

Using mathematical optimization algorithms and software in industrial environments for daily operational decision support, puts severe requirements of *robustness* on the algorithms, in the sense of solution quality, running time, and code stability. The user expects a certain time behavior and will be very irritated if one time a solution appears, for instance, after 20 seconds, another time after 12 minutes, and then again after 2 minutes. The user interface may capture very incorrect data. Beyond this, production strategies and, thus, the number and size of order widths may change over the years. This could lead to different behavior of the algorithm resulting in different running times or quality of results. Therefore, in all our implementations, we put a strong effort on optimality, strong bounds and structural independence on the ratio of typical order widths versus the widths of the master roll. This also leads to our preference for clean, simple and robust algorithms. Therefore, we have not tried to implement Branch&Price algorithms, *e.g.*, those excellent scientific pieces of work by Vanderbeck (2000) or Belov and Scheithauer (2006), which appear to us much harder to deliver in such an industrial quality.

3.6. Computational Results

We have implemented the algorithms in GAMS (v. 23.9.1) and use CPLEX (v. 12.4) to solve the resulting LP and MILP problems. The computational tests are performed on a standard laptop computer (Intel(R) i7 (dual core) with 3.3 GHz and 12.0 GB RAM) running a 64-bit Windows 7 operating system. We stop our CGA iterations when the pricing problem satisfies the stopping criteria with an absolute tolerance of 0.001. The purpose of these

numerical test runs is to demonstrate that the algorithms work and produce good or optimal results in reasonable time. Although, we did not find column generation approaches in the literature, it is not our claim that our algorithms and methods are the best or most appropriate. Therefore, our computational experiments run on a set of real world test instances with no comparisons against other authors and techniques. We rather want to present solutions to relevant real world cutting stock problems, which can be implemented in relatively short project time, which are well balanced between exact optimality and heuristics, and are easy to maintain.

The following abbreviations are used in Tables 1-3:

IT:	number of iterations (number of relaxed master problems solved)
RMP:	relaxed master problem optimal objective function value
IMP:	(integer) master problem optimal objective function value
%G:	GAP between IMP and RMP in percent, <i>i.e.</i> , $(IMP - RMP) / IMP$
DS:	demand satisfaction (E = equality, O = overproduction)
nR:	number of rolls
nP:	number of patterns
nR*:	number of rolls obtained with standard CGA
nP*:	number of patterns obtained with standard CGA
LnP:	lower bound on minimum number of patterns
MnP:	minimum number of patterns (proven)
P*:	solution uses minimal number of patterns
R*:	solution uses minimal number of rolls
T*:	solution uses minimal amount of trim-loss
B:	best solution found: the solution contains the least number of patterns among all solutions found
%Underprod:	total underproduction in percent
%W:	total waste in percent
sec.:	computational time in seconds

3.6.1. Benchmark Data Set

For the 1D CSPs, we use 25 real-world problem instances from various customers, ranging from 1 to 50 different orders. The instances are characterized by the master roll width, the number of orders, the order widths, and the number of requested pieces. The instances are available in the online supplement of “EJOR.”

3.6.2. CGA: Minimizing Underproduction

In the weighted objective function, the parameter ρ crucially effects the computed solution. The results in Table 1 reflect the delicate role of ρ . For $\rho = 0$, we obtain exact fulfillment of demand in almost all cases. Deviation, as for case 7 with one piece underfulfilled, can be explained by the difference between the solution of the restricted master problem (0.371 in this case) to the MILP master problem (84 in this case). However, the price to reach exact demand fulfillment is high, which is reflected in the large number of rolls. If we start with *large* values of ρ , minimizing the number of rolls is preferred and thus we are not surprised to see substantial underproduction. If we decrease ρ , at some value, we exactly meet demand for the first time. Usually, the number of rolls needed to fulfill demand exactly is identical to, or does not deviate too much from, the value obtained by the standard GGA. If we further decrease ρ , the number of rolls increases in most cases.

What is now the conclusion about the value of ρ ? For our problem instances, $\rho = 0.000001$ turned out to be the best value, yielding good results in almost all situations which occurred at our customer. However, this might be different in other situations. Thus, our advise is to experiment and analyze the results.

3.6.3. CGA: Master Rolls with Different Widths and Limited Availability

We modified the benchmark instances from Section 3.6.1 as follows: All data sets contain 3 master rolls of different widths. The width of the master rolls were selected as follows: roll 2 has width, B_2 , of the benchmark data set. The width of roll 1 was reduced by about 30%, $B_1 = \max\{\lfloor 0.7B_2 \rfloor, \max_i W_i\}$ allowing that the largest width can be produced, while the width of roll 3 was set to $B_3 = \lfloor 1.2B_2 \rfloor$.

Table 2 summarizes the computational results for $K = \infty$, *i.e.*, the knife constraint has not been considered, and for master rolls with different widths

and availability (equal to N for the three types); N_1 , N_2 and N_3 denote the number of rolls used. Solutions computed with the exact demand constraint are indicated by an “E” in column “DS”. If no solutions existed with the exact demand constraint or if the relative gap was larger than 20%, we allowed overproduction (“G” in column “DS”). However, in this case, the relative waste (“%W”) does not count overproduction as waste and is, thus, only of limited use.

Note that: (1) the CGA converges after a few iterations (“IT”) leading to running times of a few seconds. (2) Forcing exact demand satisfaction in the master problem (MILP) for the patterns computed leads to relatively large GAPS. Allowing overproduction results generally in smaller GAPS, but the modified round-up property does not hold (*cf.* Sect. 3.2). (3) The percentage waste (“%W”), when forcing exact demand satisfaction, is for most instances (except for “C01,” “C02,” “C03,” “C07,” “C49,” and “C50”) competitive and acceptable by practitioners. The percentage waste for the cases of overproduction excludes the overproduced pieces.

The limits, N_r , on the number of available master rolls requires us to think about the computation of an initially feasible set, \mathcal{P}_0 , of patterns. To generate \mathcal{P}_0 , we solve the unlimited CSPs for each master roll width B_r separately by the GGA. If \mathcal{P}_r^* denotes the set of patterns in the optimal solution obtained for width B_r , we obtain \mathcal{P}_0 as the union $\mathcal{P}_0 := \cup_r \mathcal{P}_r^*$. If $\sum_r N_r$ is not too small, \mathcal{P}_0 allows us to compute an initial feasible solution to the overall problem. Columns 3 to 5 in Table 2 shows that, in most cases, the larger master rolls are used up to their limits.

3.6.4. Exhaustion Method: Minimizing the Number of Rolls and Patterns

The Exhaustion Method is designed to overcome the two main drawbacks of the GGA: The number of patterns required tends to be too large and we have overproduction. Table 3 summarizes the results of the Exhaustion Method. We report on the solution computed by the GGA; GGA provides an upper bound both on the minimal number of rolls and on the minimal number of patterns. Theoretical lower bounds on the number of patterns derived from the bin packing problem (the number of rolls is not restricted) are reported in column “LnP” while the minimal number of patterns required is given in column “MnP” for the cases where the Exhaustion Method was able to prove them.

	nR*	$10^{+5}\rho$												
		0	0.001	0.01	0.1	0.5	1	5	10	20	30	40	100	1000
C01	1	5r	5r	5r	5r	1r								
C02	11	21r	11r	11r	11r	11r	11r	11r	11r	11r	11r	11r	11r	1
C03	4	14r	8r	8r	8r	8r	7r	4r	2	4r	2	4r	2	2
C04	453	1313r	468r	467r	524r	524r	527r	467r	467r	467r	467r	467r	467r	453r
C05	4	12r	5r	5r	5r	4r	4r	4r	4r	4r	3	5r	3	3
C06	17	61r	40r	1	1	1	17r	4	4	17r	4	4	46	73
C07	66	1	84r	1	1	1	1	1	1	1	1	1	1	1
C09	27	34r	27r	27r	27r	14	14	14	14	14	14	14	14	14
C10	27	37r	1	27r	27r	27r	27r	10	10	8	8	8	22	22
C11	47	60r	60r	60r	52r									
C12	30	1	32r	1	1	2	2	36	39	25	30	33	39	39
C13	44	99r	44r	44r	44r	44r	44r	44r	44r	44r	44r	44r	44r	44r
C14	7	1	6	3	6	6	6	6	6	6	6	4	3	5
C15	31	56r	33r	32r	34r	32r	31r	1	1	1	1	1	4	4
C16	19	32r	1	1	2	20	23	23	23	26	26	26	26	26
C17	7	15r	3	3	3	3	4	4	4	4	4	4	2	8
C18	97	179r	99r	99r	13	19	34	m	m	m	m	m	m	m
C19	19	8	16	11	15	24	14	13	14	15	15	14	14	13
C27	28	168r	9	9	16	8	8	14		17	12	12	28	28
C28	31	35r	32r	32r	31r	32r	32r	32r	31r	32r	31r	31r	5	13
C29	119	119r	119r	119r	119r	119r	119r	119r	119r	119r	119r	119r	119r	m
C32	21	68r	16	8	11	9	7	4	4	7	7	6	8	8
C42	33	113r	8	10	12	9	17	8	10	9	9	9	10	10
C49	440	847r	446r	445r	447r	442r	59	m	m	m	m	m	m	m
C50	453	984r	464r	3	2	1	80	m	m	m	m	m	m	m

Table 1: The number of underproduced pieces, or the number of rolls in case demand is exactly fulfilled, as a function of the weighting parameter ρ is listed. Note that in the first line we have listed $10^{+5}\rho$ instead of ρ . Entries such as 44r indicate that demand has been fulfilled exactly with 44 the number of rolls which can be compared to the number of rolls displayed in the second column (the values of the standard GGA). The flag “m” indicates runs which had underproduction with more than 100 pieces; this happens for larger values of ρ minimizing the number of rolls with significant individual wastes. In these computations we neglected the knife constraint, *i.e.*, $K = \infty$.

RC	N	N1	N2	N3	IT	RMP	IMP	%G	DS	nR	nP	nR*	nP*	%W	sec.
C01	1	0	1	0	1	405	405	0.00	E	1	1	1	1	7.92	0.26
C02	4	3	4	4	1	8109	10466	22.51	E	11	5	11	2	18.97	0.54
C03	2	2	2	0	3	714.8	910	21.45	E	4	4	4	3	10.80	0.49
C04	165	90	164	165	2	2286.3	2289.0	0.11	E	419	6	453	4	2.30	0.82
C05	2	1	1	2	3	87.95	783	87.95	E	4	4	4	4	3.69	0.77
C06	7	7	7	4	1	355.5	569.0	37.52	E	18	9	17	9	1.16	0.32
C07	24	19	24	24	1	3156.3	3260.0	3.18	E	67	10	66	9	10.62	0.28
C09	11	4	11	11	2	4383	4383	0.00	E	26	8	27	7	3.17	0.51
C10	11	5	10	11	5	2379.1	2953	19.43	E	26	9	27	6	1.77	1.68
C11	17	10	16	17	3	9140.4	10686	14.46	E	43	18	47	11	5.70	0.74
C12	11	10	8	11	3	4115.5	4318.0	4.69	E	29	12	30	8	3.37	2.71
C13	18	11	16	17	4	220	460	52.17	E	44	22	44	4	0.23	11.39
C15	12	7	12	11	2	150.6	267	43.58	E	30	17	31	16	0.33	4.67
C16	7	6	5	7	7	479.3	2346	79.57	E	18	16	31	16	0.33	4.67
C18	38	20	35	38	13	1094.9	1656	33.88	E	93	21	97	17	0.29	13.54
C28	13	3	13	10	12	225	267	15.73	E	26	19	31	18	3.84	2.57
C29	27	15	24	22	6	4656.5	4724	1.42	E	61	6	119	24	2.31	1.64
C49	177	128	177	177	1	253360	254539	0.46	E	482	38	440	32	10.48	0.25
C50	182	110	182	182	33	173297	174472	0.67	E	474	32	453	33	7.21	0.21
C02	4	3	4	4	2	8109	8467	4.22	G	11	4	11	2	15.34	0.31
C03	2	2	1	2	2	688	732	6.01	G	5	4	4	3	6.08	0.53
C05	2	1	2	2	3	75	88	14.77	G	5	4	4	4	0.25	0.79
C06	7	7	7	7	3	174.2	195.0	10.67	G	21	6	17	9	0.17	0.84
C13	18	8	17	18	5	220	224	1.78	G	43	14	44	4	0.00	1.84
C14	3	2	3	3	4	44	50	12.00	G	8	8	7	6	0.00	1.32
C15	12	7	12	12	4	109.5	115	4.76	G	31	17	31	16	0.00	1.09
C16	7	7	7	7	16	177.9	266	33.12	G	21	11	19	13	0.30	1.74
C17	3	1	3	3	17	27.2	31	12.41	G	7	7	7	7	0.00	5.50
C18	38	31	38	38	11	512.4	518	1.08	G	107	14	97	17	0.29	13.54
C19	5	4	4	1	5	67	111	39.64	G	9	9	6	6	0.02	1.37
C27	12	12	12	6	3	212	233	9.01	G	30	24	28	19	0.01	8.66
C32	8	8	8	8	7	117	135	13.33	G	24	20	21	19	0.01	7.71
C42	21	20	13	5	44	165	188	12.23	G	35	29	33	11	0.01	14.33

Table 2: CSP with three different widths for the master rolls and limited roll availability. Solutions computed with the exact demand constraint are indicated by an “E” in column “DS”. If no solutions existed with the exact demand constraint or if the relative gap was larger than 20%, we allowed overproduction (“G” in column “DS”). However, in this case, the relative waste (“%W”) does not count overproduction as waste and is, thus, only of limited use.

For the 25 real data instances tested, we observe that the Exhaustion Method finds a solution with (1) proven minimal number of rolls and proven minimal number of patterns in 10 cases (“C01,” “C02,” “C03,” “C04,” “C06,” “C13,” “C14,” “C15,” “C17,” “C19”),

(2) proven minimal number of rolls while reducing the number of patterns used by at least 1 compared to the GGA in 14 cases.

The computational times never exceed 3 minutes, even for the largest instances with 50 orders.

4. 2D trim-loss Minimization

In this section, we derive optimal patterns to satisfy demand for orders consisting of rectangles to be cut from rolls. Note that general 2D cutting (or packing) problems require completely different methods; *cf.* Vanderbeck (2001) or Lodi et al. (2002). However, in the current case we need to minimize the total trim-loss, resulting from horizontal and vertical cuts. Therefore, in our solution method we exploit the order of the cuts and solve a multi-step cutting stock problem with guillotine-cuts, which enables us to avoid the general 2D case and instead rather resort to solving the standard CSP as a subproblem. Figure 1 shows the geometry and structure of that problem. We want to cut rectangles, also called formats, from master rolls of different widths. In order to cut these rectangles, a sufficient number of sheets, whose lengths has to be determined, is cut from the selected master roll. The sheets have the same width as the master rolls, but they may not use up the whole length of the master rolls. Sheets host longitudinal stripes (longitudinal patterns), and the stripes hosts the formats (rectangles) for which we have to satisfy demand. Thus, the sequence of cuts is: vertical guillotine cuts from the master roll to produce the sheets, longitudinal guillotine cuts in the sheets, and again vertical guillotine cuts in the strips. The longitudinal patterns are subject to a practical restrictions that they should not have more than $S\%$ strip-loss.

4.1. Indices and Sets

We use the following indices:

$a \in \mathcal{A} := \{a_1, \dots, a_{N^A}\}$ the given order rectangles; $N^A \leq 25$.

The rectangles (also called, *formats*) are given

in a generic way with respect to their orientation in the longitudinal strip. We denote this by $[A_a, B_a]$, we do not distinguish the ordering, *i.e.*, $[A_a, B_a] = [B_a, A_a]$. Thus, the notation $[A_a, B_a]$ or $[B_a, A_a]$ just refers to the rectangles with sides A_a and B_a without paying attention to the specific length and width. An additional attribute signals whether the rectangles should be considered as oriented (1) or whether they could be rotated (2).

$i \in \mathcal{I} := \{i_1, \dots, i_{N^I}\}$ the given, oriented order rectangles; $N^I \leq 50$. Here, the orientation is fixed. The oriented order rectangles are obtained by duplicating the unordered order rectangles. The side A_i is measured parallel to the width W_r of a sheet; the side B_i is parallel to the longitudinal stripe. We denote this by $[A_i \times B_i]$ where the ordering is important.

$j \in \mathcal{J} := \{j_1, \dots, j_{N^J}\}$ longitudinal patterns of sheets; $N^J = 4000 - \mathcal{J}$ generated via CE. The value $N^J = 4000$ was sufficient for a particular customer case; it might be adjusted in other situations.

$r \in \mathcal{R} := \{r_1, \dots, r_{N^R}\}$ roll types; $N^R = 10$.

The master rolls are given by their width W_r and their inventory stock S_r not modeled initially.

Note that as the longitudinal patterns should not have more than $S\%$ strip-loss ($S = 2$), for a specific customer project and the size of the rectangles to be cut, N^J was sufficient, and larger than the number of longitudinal patterns.

4.2. Input Data

These are the relevant input data used:

B_r [L] the width of the master rolls.

B_{\min}, B_{\max} [cm] minimal and maximal width of the sheets; $B_{\max} = \max_r \{B_r\}$.

D_a [-] requested number of orders (demand) for order rectangle a .

I_a^R [-] indicator, whether ($I_a^R = 2$) or not ($I_a^R = 1$) order rectangle a can be rotated.

L_{\min}, L_{\max} [L] minimal and maximal length of the sheets.

M [-] the number of available knives.

	GGA		LnP	MnP	sec.	Solution(s) computed					
	nR	nP				nR	nP	P*	R*	T*	B
C01	1	1		1	3			✓	✓	✓	
C02	11	2		2	4			✓	✓	✓	
C03	4	3		2	4	4	2	✓	✓	✓	
C04	453	4		4	4			✓	✓	✓	
C05	4	4		2	5	4	3		✓		
						5	2	✓			
C06	17	9		4	10	17	4	✓	✓	✓	
C07	66	9		4	12	70	4	✓			
						66	5		✓	✓	
C09	27	7		4	13	27	4	✓	✓		
C10	27	6	5		14	33	5				
C11	47	11	8		15	51	8				
C12	30	8	6		46	30	7		✓		
						39	6	✓			✓
C13	44	4		4	16			✓	✓	✓	
C14	7	6		4	16	7	4	✓	✓	✓	
C15	31	16		7	43	31	7	✓	✓	✓	
C16	19	13	7		137	19	9		✓		
						20	8				✓
C17	7	7		5	9	7	5	✓	✓	✓	
C18	97	17	9		141	98	10				✓
						97	12		✓	✓	
C19	6	6		4	32	6	4	✓	✓	✓	
C27	28	19	5		142	27	7		✓	✓	
C28	31	18		18	24						
C29	119	24		24	5						
C32	21	19	8		150	22	9				
C42	32	27	10		157	33	11				✓
						32	12		✓	✓	
C49	440	32	12		157	446	17				
						452	16				✓
C50	453	26	10		152	471	18				✓
						459	19				
						457	20				

Table 3: Computational results for the Exhaustion Method (EM). The knife constraint has been considered. Theoretical lower bounds on the number of patterns derived from the bin packing problem (the number of rolls is not restricted) are reported in column “LnP” while the minimal number of patterns required is given in column “MnP” for the cases where the Exhaustion Method was able to prove them. For the cases “C01,” “C02,” “C03,” “C04” and “C05,” we applied the model by Johnston and Sadinlija directly to compute the minimum number, “nP,” of patterns; “nR” is the associated number of rolls - both listed in the Solution(s) computed part of the table. For each solutions of the EM, the columns “nP,” “nR,” “nT,” and “B” have the following meaning: “P*” indicates that number of patterns is minimal, “nR” indicates that the number of rolls is minimal, “T” indicates that the solution has minimal trim-loss, and “B” indicates the best solution found (not proven optimal) in case we display several solutions. If “nR” and “nP” are missing, the GGA produced a solution in which the number of patterns agrees with the proven value of “MnP.” Note that C27 is one of the cases in which the GGA produces a solution with a minimum number of rolls which can be improved by one.

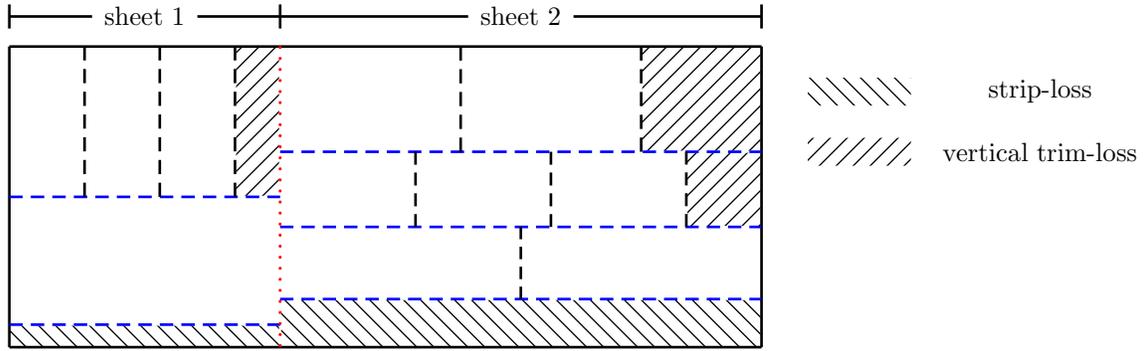


Figure 1: 2D paper cutting in three steps: 1st guillotine-cut (dotted line), 2nd longitudinal cuts, 3rd guillotine-cut (dashed vertical lines).

4.3. Variables

We use the following – mostly integer – variables:

$\mu_{rj} \in \mathbb{N}_0$ [–] states how often the sheet j can be cut from roll r ; only used in the master problem.

$\alpha_{rij} \in \mathbb{N}_0$ [–] states how often the rectangle $[A_i \times B_i]$ is contained in the sheet j of roll r . This variable is used in the subproblem and is the only independent variable of the subproblem and can be obtained explicitly.

$\ell_{rj} \in \mathbb{N}_0$ [mm] specifies the length of the sheet j of roll r ; ℓ_{rj} depends on all α_{rij} .

$w_p \geq 0$ [–] states the trim-loss of pattern p ; $0 \leq w_p \leq B$.

4.4. Overview of the Algorithmic Components

The algorithm is structured as follows:

1. The order rectangles $[A_a, B_a]$, given through their length and width, are duplicated and yield the orientated rectangles $[A_i \times B_i]$. The indices i and a are connected as follows

$$a(i) := \begin{cases} i & , \quad 1 \leq i \leq N^A \\ i - N^A & , \quad N^A + 1 \leq i \leq 2N^A \end{cases}$$

which yields

$$[A_i \times B_i] := \begin{cases} [A_{a(i)} \times B_{a(i)}] & , \quad 1 \leq i \leq N^A \\ [B_{a(i)} \times A_{a(i)}] & , \quad N^A + 1 \leq i \leq 2N^A \end{cases} .$$

or $(\forall \{i \mid 1 \leq i \leq N^A\})$

$$[B_{i'} \times A_{i'}] = [A_i \times B_i] \quad , \quad i' := i + N^A .$$

The demand for order rectangle $[A_a, B_a]$ can thus be fulfilled by orientated rectangles i and i' .

2. For each master roll r with width W_r , enumeration is used to
 - (a) generate up to N^J stripe partitions (longitudinal patterns) which are compatible to $\{A_i\}$ (the width of the stripes correspond to the width A_i) – thus, the strip-loss and the width of each longitudinal strip are known – and
 - (b) solve the corresponding subproblem (minimization of the vertical trim-loss of a stripe partition, see Section 4.5.2) – thus, the vertical trim-loss and the total trim-loss of each roll as well as its length ℓ_{rj} are known. In addition, the solution of the subproblem gives us the number, N_{raj}^{tot} , of order rectangles $[A_a, B_a]$ covered by sheet rj .
3. The master problem is a partitioning problem. This MILP problem calculates how many sheets of length ℓ_{rj} are required in order to meet the demand for order rectangles $[A_a, B_a]$.

4.5. Master- and Subproblem

4.5.1. The Master Problem: Partitioning Model

The master problem minimizes the total trim-loss, *i.e.*, the model makes use of the following objective function

$$\min \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{J}} W_{rj} \mu_{rj} \quad , \quad (29)$$

with trim-loss W_{rj} for the sheet rj ; the integer variable μ_{rj} denotes its multiplicity. The demand equa-

tion reads

$$D_a \leq \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{J}} N_{raj}^{\text{tot}} \mu_{rj} \leq D_a + D_a^{\text{over}} \quad , \quad \forall a \quad (30)$$

or

$$D_i \leq \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{J}} N_{rij}^{\text{tot}} \mu_{rj} + \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{J}} N_{r,i+N^A,j}^{\text{tot}} \mu_{rj} \leq D_i + D_i^{\text{over}} \quad , \quad \forall \{i \mid 1 \leq i \leq N^A\} \quad . \quad (31)$$

The multiplicity variables can be bounded by

$$\mu_{rj} \leq \max_a \left\{ \left\lceil \frac{D_a + D_a^{\text{over}}}{\max\{1, N_{raj}^{\text{tot}}\}} \right\rceil \right\} \quad , \quad \forall \{rj\} \quad . \quad (32)$$

The integrality constraints are given by

$$\mu_{rj} \in \mathbb{N}_0 \quad , \quad \forall \{rj\} \quad . \quad (33)$$

For operative reasons, it might be useful to use only rolls of a particular width r_* , because several of such rolls might be cut on top of each other simultaneously (*e.g.*, up to four rolls can be cut at the same time in our case). In this case, model (29)-(33) is substituted by a sequence of N^R models, where each time only one of the N^R roll widths is used; the user can then select the appropriate solution. Care has to be taken when restricting overproduction; this can lead to infeasibilities when considering a single roll width. It might be useful to consider a goal programming approach which first minimizes the overproduction and second the trim-loss, or vice versa.

Alternatively, we introduce the binary variables δ_r which is 1 if a roll of width r is used and 0 otherwise. The following equation

$$\sum_{r \in \mathcal{R}} \delta_r = 1$$

and inequalities

$$\mu_{rj} \leq \delta_r \quad , \quad \forall \{rj\}$$

need to be added to the model.

The inventory stock can be modeled with the following approach: Each individual master roll t by its roll type r (width) and its (remaining) length L_t . Then, the following constraints

$$\mu_{rj} = \sum_{t \mid R_t = \text{ord}(r)} \mu_{rjt} \quad , \quad \forall \{rj\}$$

and

$$\sum_{t \mid R_t = \text{ord}(r)} L_{rj} \mu_{rjt} \leq L_t \quad , \quad \forall \{rjt\}$$

need to be satisfied. The integer variable μ_{rjt} describes, how many sheets of type rj are to be cut from master roll t .

4.5.2. The Subproblem

For a given distribution of longitudinal cuts for a sheet rj – *i.e.*, a system of values A_i , $1 \leq i \leq 2N^A$ – the task is to decide on the lengths ℓ_{rj} of the sheet such that the vertical trim-loss w_{rj}^{ver} , the total trim-loss w_{rj}^{tot} , or the w_{rj}^{rel} striploss is minimized. The integer variable α_{rij} defines the amount of times the oriented rectangle $[A_i \times B_i]$ is contained in the longitudinal stripe i (defined through A_i) with multiplicity N_{rij}^{pat} . This enables us to express the vertical trim-loss as

$$w_{rj}^{\text{ver}} := \sum_{i \mid N_{rij}^{\text{pat}} > 0} (N_{rij}^{\text{pat}} A_i) s_{rij} \quad ,$$

$$s_{rij} := \ell_{rj} - B_i \alpha_{rij} \quad .$$

The absolute and relative trim-loss – the latter leading to a MINLP problem – are given through

$$w_{rj}^{\text{tot}} := W_r \ell_{rj} + w_{rj}^{\text{ver}}$$

and

$$w_{rj}^{\text{rel}} := W_r + w_{rj}^{\text{ver}} / \ell_{rj} \quad .$$

The relative trim-loss measure for the objective function leads to more large sheets close to the limit L_{max} . However, they will contain more formats.

The model is restricted by

$$L_{\text{min}} \leq \ell_{rj} \leq L_{\text{max}}$$

and ($\forall \{rij \mid N_{rij}^{\text{pat}} = 0\}$)

$$\alpha_{rij} = 0 \quad ,$$

$$\alpha_{rij}^{\text{min}} := \left\lceil \frac{L_{\text{min}}}{B_i} \right\rceil \leq \alpha_{rij} \leq \left\lfloor \frac{L_{\text{max}}}{B_i} \right\rfloor =: \alpha_{rij}^{\text{max}} \quad .$$

4.5.3. Explicit Solution of the Subproblems

For computational efficiency, we suggest to calculate the solution of the subproblems and all derived data (α_{rij} , ℓ_{rj}) simultaneously with CE. The structure of the subproblem allows us to compute the optimal values of α_{rij} analytically. The flowchart for the explicit calculation of α_{rij} for sheet rj is summarized in Algorithm 4.1.

Note that this explicit solution method works for the absolute as well as the relative trim-loss.

Algorithm 4.1 Explicit solution of the subproblem

```
1:  $z_{\min} := +\infty$  // Initialization of the trim-loss
2: for all  $i = 1, \dots, N^W = 2N^A$  do
3:   if  $N_{ij}^{\text{pat}} = 0$  goto line 26
4:    $\alpha_i^{\min} := \left\lceil \frac{L_{\min}}{B_i} \right\rceil$ ,  $\alpha_i^{\max} := \left\lfloor \frac{L_{\max}}{B_i} \right\rfloor$ 
5:   // loop over all feasible  $\alpha_i$ 
6:   for all  $k = \max\{1, \alpha_i^{\min}\}, \dots, \alpha_i^{\max}$  do
7:      $\alpha_i := k$ ,  $\ell := B_i k$ ,  $w^{\text{strip}} := W\ell$ ,  $w^{\text{ver}} := 0$ 
8:     // inner loop over all  $i' \neq i$ 
9:     for all  $i' = 1, \dots, 2N^A$  do
10:      if  $N_{ri'j}^{\text{pat}} = 0$  or  $i' = i$  goto line 15
11:       $\alpha_{i'} := \left\lfloor \frac{\ell}{B_{i'}} \right\rfloor$ ,  $w_{i'}^{\text{ver}} := N_{ij}^{\text{pat}} A_i (\ell - B_{i'} \alpha_{i'})$ ,  $w^{\text{ver}} := w^{\text{ver}} + w_{i'}^{\text{ver}}$ 
12:    end for
13:     $z := w^{\text{strip}} + w^{\text{ver}}$  // absolute trim-loss
14:     $z := \frac{w^{\text{strip}} + w^{\text{ver}}}{\ell}$  // relative trim-loss
15:    // check, whether or not  $z$  leads to an improvement compared to  $z_{\min}$ 
16:    if  $z < z_{\min}$  then
17:       $\ell_* := \ell$ ,  $w_*^{\text{strip}} := w^{\text{strip}}$ ,  $w_*^{\text{ver}} := w^{\text{ver}}$ 
18:      for all  $i'' = 1, \dots, 2N^A \mid N_{i''j}^{\text{pat}} > 0$  do
19:         $\alpha_{i''}^* := \alpha_{i''}$ ,  $w_{i''}^{\text{ver}} := N_{ri''j}^{\text{pat}} A_{i''} \left( \ell_* - B_{i''} \left\lfloor \frac{\ell_*}{B_{i''}} \right\rfloor \right)$ 
20:      end for
21:    end if
22:  end for
23: end for
24: return  $\alpha^*$  and  $\ell_*$ 
```

4.5.4. Assignment of Sheets to the Orders

If rolls and sheets are treated as individuals, a result such as

Order A25022007 is satisfied by:
 Sheet 1-433 of roll R102-stock0017
 Sheet 22-40 of roll R152-stock0002

enables us to know how the orders are satisfied. Further, we might be interested what happens with each rectangle cut from a roll and sheet, *i.e.*, for which order it is intended.

The assignment of rectangles in the sheets to the original orders is done after optimizing the trim-loss. The algorithm does not require any additional data or results other than the ones provided for the trim-loss minimization problems. However, the following derived data are necessary:

$i = i(a)$ [-] rectangle i , corresponding to order a [derived entry date].

$i' = i'(a)$ [-] rectangle i' , obtained through rotation of rectangle i and corresponding to order a [derived entry date].

D_a [-] the order quantity for order a [derived entry date].

N_{rji} [-] the quantity how often rectangle i is contained in sheet rj .

$N_{rji'}$ [-] the quantity, how often rectangle i associated with rectangle i' is contained in sheet rj .

μ_{rj} [-] the quantity, how often sheet rj is used.

$X_{rja} := \mu_{rj}(N_{rji} + N_{rji'})$ [-] the quantity, how often the order a corresponding to rectangles i and i' are produced by sheet rj .

The model and algorithm requires the following variables:

$x_{arj} \in \mathbb{N}$ [-] the number of rectangles, cut from the sheet rj , used for order a .

The variables can assume values between 0 and $X_{rj}(N_{rji} + N_{rji'})$.

$y_{arj} \in \{0, 1\}$ [-] indicates, whether (1) or not (0) order a is served by sheet rj .

The goal is to minimize the sum over all quantities $s_a := \sum_{rj} y_{arj}$, which measures how many sheets serve order a , *i.e.*,

$$\min \sum_{a \in \mathcal{A}} s_a = \sum_{a \in \mathcal{A}} \sum_{rj} y_{arj} \quad .$$

The demand satisfaction is a constraint

$$\min \sum_{rj} x_{arj} = D_a \quad .$$

The variables x_{arj} are restricted by

$$x_{arj} \leq \min\{X_{arj}, D_a\} y_{arj} \quad ; \quad \forall \{arj\} \quad .$$

Note: This assignment problem is solved *after* the minimization problem. The quality of the results with respect to trim-loss and the quantity of sheets is *not* affected.

4.6. Computational Results

We use the same computational framework as for the 1D case (*cf.* Sect. 3.6). The details of the 12 test instances used are summarized in Table 4. Each 2D instance is characterized by the number of orders, N^A , and the size of the rectangles to be cut (*i.e.*, length and width), quantity of requested pieces, number of different-size master rolls, N^R , and their widths. Each master roll is assumed to have infinite length.

The results for the 2D test instances are summarized in Table 5. For the reported optimal solution, N^S is the number of different sheets required, “S” (surplus) indicates the number of overproduced pieces, “%W” is the relative waste in percent and “T” shows the time in milliseconds. The interpretation of the “Optimal Solution” is illustrated by the example $1,213 \times [4A2 \cdot 3 + A1r \cdot 5] \rightarrow 2$: The “ $\rightarrow 2$ ” indicates that this sheet is cut from roll type 2 with multiplicity 1,213. “4A2·3” expresses that order A2 enters $N_{rij}^{\text{pat}} = 3$ times and that $\alpha_{rij} = 4$; the total contribution of that sheet to order A2 is thus $1,213 \cdot 4 \cdot 3 = 14,556$ pieces. The term “A1r·5” indicates that the sheet contains order A1 in rotated placement and contributes $1,213 \cdot 1 \cdot 5 = 6,065$ pieces.

The table shows that the model and algorithm can handle various different situations regarding size of the rectangles and from small demand numbers to large ones reaching up to 1,000,000. All results were obtained in fractions of one second. From a practical point, sheets which are used only once or with small multiplicity, are not desirable. As they do not contribute many pieces, they are usually neglected – little underproduction is not a problem.

	Orders			Master Roll			
	N ^A	Dimensions [mm]	Quantity	N ^R	Width [mm]		
2D01	2	21×22, 25×27	20,812 21,367	3	102	122	152
2D02	3	36×80, 26×85, 24×33	8,000 8,000 20,000	2	102	152	
2D03	9	24×36, 36×80, 29×100 39×103, 29×100, 39×93 19×75, 29×68, 19×29	20,000 15,000 5,000 5,000 5,000 5,000 5,000 15,000 15,000	3	152	122	102
2D04	1	101×43.5	10,500	1	127		
2D05	3	48×110, 43.5×101, 37.5×87	15,000 15,000 20,000	1	122		
2D06	1	3×2 “Schnipselroll”	1,000,000	1	102		
2D07	6	43.5×101, 33.5×79, 28×68 23×58, 48×110, 53×122	10,500 70,500 43,000 21,500 2,000 4,500	2	127	122	
2D08	1	3×2 “Schnipselroll”	1,000,000	1	102		
2D09	2	as 1 but 85% striploss,	100 pc overproduction	3	102	122	152
2D10	6	as 7 but 82% striploss	as 7	2	127	122	
2D11	1	as C08	10 pc overproduction	1	102		
2D12	1	50×50	5556	8	102	112	122 152
					162	182	203 228

Table 4: Data for the 2D test instances.

	N ^S	Optimal Solution	S	%W	T	
2D01	6	1×[3A2·2+A2r·2]→1 , 2×[4A2·4+A1r·5]→2 290×[6A1·5+A1·4]→3 , 1×[6A2·2]→3 1×[3A1·6+3A1·5]→3 , 1,007×[4A2·6+A1r·5]→3	0	0.65	78	
2D02	5	1×[3A2·1+A3·2]→1 , 1,996×[3A2·2]→1 , 1×[2A2·1+3A3·4]→2 2,665×[2A1·1+3A2·2]→2 , 1,335×[1A1r·1+2A3r·3]→2	0	3.87	109	
2D03	9	625×[8A7·1]→1 , 1,000×[A2·1+2A2r·1]→1 2,750×[2A2r·4+A8r·5]→1 , 1,000×[A9·5+A1·6+A3·5]→1 1,000×[A9·5+A1·6+A5·5]→1 , 50×[A1·2]→2 , 2,500×[A1·3+A4·1]→3 2,500×[A6r·1+A9r·2]→2 , 1,250×[A8·1+A9r·2]→2	0	1.84	125	
2D04	1	10,500×[A1r·1]→1	0	20.47	47	
2D05	3	10,000×[1A2·1+2A3·1]→1 , 5,000×[A2r·1]→1 , 15,000×[A1r·1]→1	0	11.47	47	
2D06	1	258×[102A1·38]→1	8	0	50	
2D07	8	1×[2A2·1+2A3·1]→1 , 8,350×[3A2·1+A3·1]→1 , 1,852×[2A2·2+A4r·7]→1 , 4,777×[2A3·2+A3r·5]→1 , 1×[2A3r·5+A4r·6]→1 , 10,500×[A1·1+A2r·3]→1 , 2,000×[A5·1+A2r·3]→1 , 4,500×[A6·1]→2	0	4.30	47	
2D08	1	1,548×[34A1·19]→1	8	0	47	
2D09	2	1,213×[4A2·4+A1r·5]→2 , 492×[6A1·5+A2·4]→3	3	9	0.66	62
2D10	8	6,382×[2A2·1+2A3·1]→1 , 17,494×[3A2·1+A4·1]→1 , 2,000×[A4·2+A5·1+A6·1]→1 , 7,559×[4A3·1]→2 , 5,250×[2A1·5+A2·6]→2 , 1×[3A4·1+A6·3]→2 , 1×[2A2·1+A6·3]→2 , 1,249×[2A6·1]→2	0	8.08	62	
2D11	1	as C08	8	0	47	
2D12	1	1,852×[3A1·1]→4	0	1.32	312	

Table 5: Optimal solutions computed by the polyolithic solution method for the test instances of Table 4.

5. Current-Edge Cutting Stock Problems

We assemble various CSPs which are at the forefront of research or cannot be found in the literature.

5.1. CSPs Under Uncertainty

An important practical challenge is how to deal with variations in demand. We follow the spirit by Beraldi et al. (2009) and treat the demand stochastically: In the first stage, decisions on the choice of patterns have to be made to satisfy first stage demand (D_i) while their multiplicity can be adjusted in a second-stage when the stochastic second-stage demand (D_{is}) unfolds. The task is then to minimize the total expected number of patterns used.

The column enumeration procedure (*cf.* Sect. 3.3) can be adjusted in a straight forward manner to deal with demand uncertainty. Once the columns (collected in set \mathcal{P}') have been enumerated, we solve the following MILP:

$$\min \sum_{p \in \mathcal{P}'} \left(\mu_p + \sum_{s \in \mathcal{S}} p_s \mu_{ps} \right) ,$$

with probability p_s of scenario s occurring, subject to the first-stage demand-fulfill inequalities (8), the second-stage demand constraints

$$\sum_{p \in \mathcal{P}'} N_{ip} \mu_{ps} \geq D_{is} \quad , \quad \forall \{is\} \quad ,$$

and the connection of the first-stage and second-stage pattern

$$\delta_p \leq \mu_p \leq M_p \delta_p \quad , \quad \forall p \in \mathcal{P}'$$

and

$$\mu_{ps} \leq M_p \delta_p \quad , \quad \forall p \in \mathcal{P}' \quad , \quad \forall s \quad ,$$

where binary variable δ_p indicates whether or not patten p is used in the first stage or not and M_p is a sufficiently large constant. We require integrality on the decision variables

$$\mu_{ps} \in \mathbb{N}_0 \quad , \quad \forall p \in \mathcal{P}' \quad , \quad \forall s$$

and

$$\delta_p \in \{0, 1\} \quad , \quad \forall p \in \mathcal{P}' \quad ,$$

as well as integrality constraint (9).

Alem et al. (2010) also consider stochastic demand for CSPs. However, the first stage decision is on the multiplicity and pattern choice where

the recourse decisions determine over- or under-production of the stochastic demand realized. The objective is then to minimize the cutting cost plus penalty cost for over- and under-production. The resulting problem structure allows us to apply a column generation approach.

5.2. CSPs with Tolerances

In the literature, cutting stock problems appear with certain number of order rolls demanded. However, in reality, the orders are often specified in weight. If one has master rolls of different lengths (!) available, the total weight ordered is subject to different number of rolls cut to a certain pattern. Both, the ordered weights and widths, are subject to tolerances.

5.3. CSPs with Limited Inventory

In Section 3.5.5 we have provided a CGA for the simplest case of master rolls with different widths and limited stock availability, but under the assumption that all have the same length (usually, infinity). However, *on stock* may imply, that the rolls on stock have also finite and different lengths. This case may occur under certain production philosophies and forces us to treat rolls as individuals.

5.4. CSPs and Simultaneous Production Planning and/or Scheduling

These problems are really hard as we have to deliver at certain due dates but there are already examples in the literature as outlined in Section 2. Here we find *make-to-stock* versus *make-to-order* (and mixed strategies). This class of problems is challenging as cutting procedures and production scheduling vary from case to case.

6. Paper Industry and Optimization

In recent times, the paper industry has vigorously cut back their capacity, rising the prices. Rising energy and raw material costs pressure the manufacturers. Many machines were shut down worldwide. Also, the concentration process has continued.

6.1. Optimization Software: Importance and Financial Impact

The use of automated, software-based waste optimization provides great benefits to all producers of rolls or formats such as paper mills, film producers and steelmakers.

A manually performed cut plan by a production planner, even though s/he might have significant experience and knowledge, is, depending on the complexity, very time consuming and reaches, as opposed to a software-based plan, usually not the optimum. This becomes even more significant for producers with very different widths or format sizes and different machines.

Our customers use the trim optimization both when determining the optimal width of the parent rolls and when planning secondary production processes. Waste optimization, integrated into the planning process, is an essential, indispensable element of the short- and long-term production planning. Depending on the complexity of the process, using the integrated waste optimization, even an automated production planning is possible, *e.g.*, a weekly schedule.

For paper factories, which produce only standard or simple cutting patterns, mathematical optimization may not be beneficial. Manual results are hard to beat for plants with very standardized product portfolio. In contrast, for plants which encounter a broad spectrum of products, our experience is as follows. Depending on the complexity of their production structure, our customers confirmed 3-6 % reduced trim-loss, comparing mathematical optimization based cutting stock solutions to their manual procedures.

Let us inspect this in detail for a typical customer producing special paper: For an annual production of 50,000 tons a trim-loss reduction of 3% saves 1,500 tons per year. Even if the trim loss is used in a recycling mill as raw material, the total saving is 285,000 Euro production costs at the paper machine if we apply typical production cost of 190 Euro per ton of produced paper, not considering the raw material price. Those paper factories which cannot recycle the savings are significantly higher. The one-time purchase cost for our computer-based solution is about 30,000 to 40,000 Euro. Under the conditions of this example, the Return of Investment (ROI) is reached after only two months. A further advantage is the decrease of dependence on employees with many years of knowledge and experience in the creation of manual cutting plans.

On a global scale, the financial benefit of optimization tools in the cutting paper industry varies, *e.g.*, dependent on the machine type and order variety. In our experience, manual plans exhibit an average waste of between 3-5%, while optimization yields in less than 2% waste, often below 1%. One

percentage waste minimization may lead to an annual saving of one million Euro per paper machine.

6.2. Outlook

In our experience, the diversification of our clients' products and the industry as a whole increases steadily, so that the use of a software-supported, automated production planning with integrated cutting stock optimization becomes more and more a necessity. Solving cutting stock problems is not an isolated activity any longer.

7. Conclusions

Motivated by challenges arising in the paper industry, we have presented various polyhedral solution approaches to model and solve a testbed of 1D and 2D real-world cutting problems. The methods are implemented in GSE's ERP system. We have developed new column generation approaches to treat problems with different master roll widths, limited number of master rolls on stock, or problems in which underproduction is allowed but to be minimized. Numerical experiments demonstrate the value of these approaches and in some data instances also suggest either the use of Branch&Price techniques or other solution methods like the Exhaustion Method, which is very efficient and produces near optimal solutions for minimizing the number of patterns. The presented polyhedral solution methods yield practically superior solutions compared to the standard GGA, with practically feasible computational times.

As each paper producing company has its own production processes specific cutting machines, the mapping of reality and abstraction into a mathematical model and algorithm has, to our experience, the highest priority to receive the appreciation of our clients. Besides this, future research and efficient mathematical techniques on the integration of CSPs into the production planning process and the incorporation of uncertainty would be very welcome.

Appendix A. Post-Processing of Optimal GGA/CGA Solutions

The GGA/CGA produces solutions which are minimal in the number of rolls but are occasionally lead to overproduction exceeding the demand

D_i by S_i pieces. We have developed three heuristics to eliminate overproduction: The first one just modifies patterns, the second one modifies patterns and generates new ones, and the third one is a combination of both.

In *Case 1*, for a given S_i , we can find a pattern p used with multiplicity M_p and $N_{ip} > 0$ subject to

$$\exists \kappa \mid S_i = \kappa M_p \quad \wedge \quad \kappa = 1, 2, \dots, N_{ip} \in \mathbb{N} \quad , \quad (\text{A.1})$$

where κ is an integer number. In this simple case, we modify pattern p as follows:

$$N_{ip} := N_{ip} - \kappa \quad \text{and} \quad w_p := w_p + \kappa W_i \quad .$$

This implies that the number of pieces i , produced by using pattern p M_p times, is reduced by $S_i = \kappa M_p$.

Case 2 is more complicated. Let $R_{ip} := N_{ip} M_p$ denote the number of pieces i produced by using pattern p M_p times. If we allow a new pattern p' , in which item i occurs μ times, to be used π times, the reduction condition reads now

$$(N_{ip} - \kappa)(M_p - \pi) + \mu\pi = R_{ip} - S_i$$

or

$$(\kappa + \mu - N_{ip})\pi = \kappa M_p - S_i \quad , \quad (\text{A.2})$$

where π reduces the number of rolls cut to pattern p . The term $\mu\pi$ considers the number of pieces of type i produced by using the new pattern π times. Depending on the solution of (A.2), p' does not contain item i at all ($N_{p'i} = 0$) or has a reduced number of it ($N_{p'i} < N_{ip}$) while others items i' remain unchanged ($N_{i'p'} = N_{ip}$). For $\pi = 0$ we obtain the simple case (A.1) above; it suffices to modify just one pattern.

We solve the diophantine equation (A.2) by tests in three nested loops:

$$\begin{aligned} \pi &= 0, 1, \dots \quad , \\ k &= 1, 2, \dots, N_{pi} \quad , \\ \mu &= 0, 1, \dots \quad . \end{aligned}$$

The existing pattern p is modified by

$$N_{ip} \rightarrow N_{ip} - \kappa \quad \text{and} \quad M_p \rightarrow M_p - \pi \quad .$$

For p' , item i and all other items $i' \neq i$ we apply

$$\begin{aligned} N_{i'p'} &\rightarrow N_{ip} \quad , \\ N_{ip'} &\rightarrow \mu \quad , \\ M_{p'} &\rightarrow \pi \quad . \end{aligned}$$

Unfortunately, there are still situations not covered by Case 1 and Case 2. Let us illustrate this for item $i = 14$ (also called A14) by the following example with $D_{14} = 1$:

	item	N_{ip}	M_p
p11	A14	1	2
p12	A14	1	5
p23	A14	1	1

We have $S_{14} = 7$ surplus items. The patterns p_{11} , p_{12} and p_{23} contain other items not relevant here. The equality (A.1) $S_i = \kappa M_p$ does not have a solution for $\kappa \leq 1$; similarly, (A.2) does not have a solution. The best we could do is to eliminate A14 in p_{23} which leaves a surplus of $S'_{14} = 6$.

Therefore, we add *Case 3*. For those widths i with $S_i > 0$ we generate a list $L_i := \{p'_1, p'_2, p'_3, \dots\}$, which contains those patterns in which item i occurs, *i.e.*, for all elements $p'_m \in L_i$ we have: $N_{ip_m} > 0$ with $p_m = p(p'_m)$ for all m . Now we check for all m , whether $\kappa \geq 1$ exists with $\kappa M_p \leq S_i$. If it exists, item i in pattern p_m is eliminated k times and we obtain $S'_i := S_i - \kappa M_p$. If $S'_i > 0$, we continue the test for the next m and pattern p_m until all patterns in L_i have been tested. If after this, we still have $S'_i > 0$, we continue with *Case 2*.

Appendix B. Deriving the Pricing Problem

In this section, we briefly describe how the pricing problems of the cutting stock problem are derived.

Let the master problem (minimization problem) contain all possible patterns \mathcal{P} ; the *restricted* master problem the patterns $\mathcal{P}' \subseteq \mathcal{P}$. An optimal solution of restricted master problem defines also an optimal solution for the master problem, if all the reduced cost

$$c_p - \pi A_p \geq 0 \quad \forall p \in \mathcal{P} \quad (\text{B.1})$$

with objective function coefficient c_p for variable μ_p , (vector of) dual variables π of all functional constraints and A_p being the constraint coefficient matrix of variable μ_p .

Thus the pricing problem is obtained by minimizing (B.1) over all p . Naturally, the constraints of the pricing problem ensure that the newly computed pattern is feasible, *i.e.*, is part of the set \mathcal{P} .

Let us illustrate this by the objective function term

$$\sum_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}'_r} \left(B_r - \sum_{i \in \mathcal{I}} (W_i - 1) N_{ip} \right) \mu_p \quad (\text{B.2})$$

occurring in (27). From this term and the inequalities (28)

$$\sum_{p \in \mathcal{P}'_r} \mu_p \leq N_r \quad , \quad \forall r \quad , \quad (\text{B.3})$$

and (8)

$$\sum_{p \in \mathcal{P}'_i} N_{ip} \mu_p \geq D_i \quad , \quad \forall i \quad , \quad (\text{B.4})$$

we had derived the pricing problem

$$z_r^* := \min \left(B_r - \sum_{i \in \mathcal{I}} (W_i - 1) \alpha_i - \pi_r - \sum_{i \in \mathcal{I}} \pi_i \alpha_i \right) . \quad (\text{B.5})$$

We note in (B.5) that the $\sum_{p \in \mathcal{P}'_r}$ has disappeared, and that all occurrences of N_{ip} have been replaced by α_i . Instead of $\sum_{r \in \mathcal{R}}$ we have to deal with r independent pricing problems to be solved. The coefficients appearing with the Lagrange multipliers π_r and π_i are those in (28) and (8) appearing with the μ_p variables.

References

- Alem, Jr., D., Munari, Jr., P., Arenales, M., Ferreira, P., 2010. On the cutting stock problem under stochastic demand. *Annals of Operations Research* 179, 169–186.
- Alves, C., de Carvalho, J. M. V., 2007. Accelerating column generation for variable sized bin-packing problems. *European Journal of Operational Research* 183 (3), 1333–1352.
- Alves, C., Macedo, R., de Carvalho, J., 2009. New Lower Bounds Based on Column Generation and Constraint Programming for the Pattern Minimization Problem. *Computers & Operations Research* 36, 2944–2954.
- Amor, Hatem Ben de Carvalho, J. M. V., 2005. Cutting Stock Problems. In: Desrosiers, G. D. J., Solomon, M. M. (Eds.), *Column Generation*. Springer, New York, pp. 131–161.
- Arbib, C., Marinelli, F., 2005. Integrating process optimization and inventory planning in cutting-stock with skiving option: An optimization model and its application. *European Journal of Operational Research* 163 (3), 617–630.
- 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.
- Beraldi, P., Bruni, M., Conforti, D., 2009. The stochastic trim-loss problem. *European Journal of Operational Research* 197, 42–49.
- Cui, Y., Zhao, Z., 2013. Heuristic for the rectangular two-dimensional single stock size cutting stock problem with two-staged patterns. *European Journal of Operational Research* 231, 288–298.
- Dyckhoff, H., Jan. 1990. A Typology of Cutting and Packing Problems. *European Journal of Operational Research* 44 (2), 145–159.
- Gilmore, P., Gomory, R., 1961. A Linear Programming Approach to the Cutting Stock Problem. *Operations Research* 9, 849–859.
- Gramani, M. C. N., França, P. M., 2006. The combined cutting stock and lot-sizing problem in industrial processes. *European Journal of Operational Research* 174 (1), 509–521.
- Gramani, M. C. N., França, P. M., Arenales, M. N., 2009. A Lagrangian Relaxation Approach to a Coupled Lot-sizing and Cutting Stock Problem. *International Journal of Production Economics* 119, 219–227.
- Haessler, R. W., 1971. A Heuristic Programming Solution to a Nonlinear Cutting Stock Problem. *Management Science* 17, B793–B802.
- Haessler, R. W., 1992. One-dimensional cutting stock problems and solution procedures. *Math. Comput. Model.* 16 (1), 1–8.
- Haessler, R. W., Sweeney, P. E., 1991. Cutting Stock Problems and Solution Procedures. *European Journal of Operational Research* 54, 141–150.
- Haessler, R. W., Talbot, F. B., 1983. A 0-1 Model for Solving the Corrugator Trim Problem. *Management Science* 29, 200–209.
- Harjunkoski, I., Westerlund, T., Pörn, R., Skrifvars, H., 1998. Different transformations for solving non-convex by MINLP. *European Journal of Operational Research* 105, 594–603.
- Holthaus, O., 2002. Decomposition approaches for solving the integer one-dimensional cutting stock problem with different types of standard lengths. *European Journal of Operational Research* 141 (2), 295–312.
- Johnston, R., Sadinlija, E., 2004. A New Model for Complete Solutions to One-Dimensional Cutting Stock Problems. *European Journal of Operational Research* 153, 176–183.
- Kallrath, J., 2009a. Combined Strategic Design and Operative Planning in the Process Industry. *Computers and Chemical Engineering* 33, 1983–1993.
- Kallrath, J., 2009b. Cutting Circles and Polygons from Area-Minimizing Rectangles. *Journal of Global Optimization* 43, 299–328.
- Kallrath, J., 2011. Polyhedral Modeling and Solution Approaches Using Algebraic Modeling Systems. *Optimization Letters* 5, 453–466.
- Kallrath, J., Rebennack, S., 2014. Cutting Ellipses from Area-Minimizing Rectangles. *Journal of Global Optimization*, to appear.
- Li, S., 1996. Multi-Job Cutting Stock Problem with Due Dates and Release Dates. *The Journal of the Operational Research Society* 47 (4), 490–510.
- Lodi, A., Martello, S., Monaci, M., 2002. Two-dimensional packing problems: A survey. *European Journal of Operational Research* 141 (2), 241–252.
- McDiarmid, C., 1999. Pattern Minimisation in Cutting Stock Problems. *Discrete Applied Mathematics* 98, 121–130.
- Poldi, K. C., Arenales, M. N., 2009. Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Computers & OR* 36 (6), 2074–2081.
- Poltroniere, S. C., Poldi, K. C., Toledo, F. M. B., Arenales, M. N., 2008. A coupling cutting stock-lot sizing problem in the paper industry. *Annals OR*, 91–104.
- Pörn, R., Harjunkoski, I., Westerlund, T., 1999. Convexification of different classes of non-convex MINLP problems. *Computers & Chemical Engineering* 23, 439–448.
- Rebennack, S., Kallrath, J., Pardalos, P. M., 2009. Column Enumeration based Decomposition Techniques for a Class of Non-convex MINLP Problems. *Journal of Global Optimization* 43, 277–297.
- Reinertsen, H., Vossen, T. W. M., 2010. The one-dimensional

- Cutting Stock Problem with Due Dates. *European Journal of Operational Research*, 701–711.
- Rodríguez, M. A., Vecchietti, A., 2008. Enterprise optimization for solving an assignment and trim-loss non-convex problem. *Computers and Chemical Engineering* 32, 2812–2822.
- Santos, M. O., Almada-Lobo, B., Aug. 2012. Integrated pulp and paper mill planning and scheduling. *Comput. Ind. Eng.* 63 (1), 1–12.
- Scheithauer, G., Terno, J., 1995. The modified integer round-up property of the one-dimensional cutting stock problem. *European Journal of Operational Research* 84, 562–571.
- Trkman, P., Erjavec, J., Gradisar, M., 2009. Cutting as a Continuous Business Process. *Economic and Business Review* 11 (4), 285–300.
- Trkman, P., Gradisar, M., 2007. One-dimensional cutting stock optimization in consecutive time periods. *European Journal of Operational Research* 179 (2), 291–301.
- Vanderbeck, F., 2000. Exact Algorithm for Minimising the Number of Setups in the One-dimensional Cutting Stock Problem. *Operations Research* 48, 915–926.
- Vanderbeck, F., 2001. A nested decomposition approach to a three-stage, two-dimensional cutting-stock problem. *Management Science* 47 (6), 864–879.
- Wäscher, G., Haubner, H., Schumann, H., Dec. 2007. An improved typology of cutting and packing problems. *European Journal of Operational Research* 183 (3), 1109–1130.
- Yanasse, H. H., Pinto Lamosa, M. J., 2007. An integrated cutting stock and sequencing problem. *European Journal of Operational Research* 183 (3), 1353–1370.