Models for two- and three-stage two-dimensional cutting stock problems with a limited number of open stacks

Mateus Martin[a,b,*], Horacio Hideki Yanasse[b], Maristela O. Santos[c], Reinaldo Morabito[d]

[a] *Departamento de Engenharia de Produção, Universidade Federal Fluminense, Rua Domingos Silvério 135, Petrópolis-RJ, 25650-050, Brazil*

[b] *Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, Avenida Cesare Mansueto Giulio Lattes 1201, 12247-014, São José dos Campos-SP, Brazil*

[c] *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Avenida Trabalhador São-carlense 400, 13566-590, São Carlos-SP, Brazil*

[d] *Departamento de Engenharia de Produção, Universidade Federal de São Carlos, Via Washington Luiz km 235, 13565-905, São Carlos-SP, Brazil*

Abstract

We address three variants of the two-dimensional cutting stock problem in which the guillotine cutting of large objects produces a set of demanded items. The characteristics of the variants are: the rectangular shape of the objects and items; the number of two or three orthogonal guillotine stages; and, a sequencing constraint that limits the number of open stacks to a scalar associated with the number of automatic compartments or available space near the cutting machine. These problems arise in manufacturing environments that seek minimum waste solutions with limited levels of work-in-process. Despite their practical relevance, we are not aware of mathematical models for them. In this paper, we propose an integer linear programming (ILP) formulation for each of these variants based on modeling strategies for the two-dimensional guillotine cutting stock problem and the minimization of open stacks problem. The first two variants deal with exact and non-exact 2-stage patterns, and the third with a specific type of 3-stage patterns. Using a general-purpose ILP solver, we performed computational experiments to evaluate these approaches with benchmark instances. The results show that the the several equivalent solutions of the cutting problem allows obtaining satisfactory waste solutions with a reduced number of open stacks.

Keywords: Cutting stock problems, Mixed-integer linear programming, Pattern sequencing, Integrated problems

---

*Corresponding author.

Email addresses: mpmartin@id.uff.br (Mateus Martin), horacio.yanasse@unifesp.br (Horacio Hideki Yanasse), mari@icmc.usp.br (Maristela O. Santos), morabito@dep.ufscar.br (Reinaldo Morabito)

1. Introduction

The two-dimensional cutting stock problem (2D-CSP) deals with the cutting of a set of rectangular item types with a pre-determined length, width, and demand out of a minimum number of rectangular large objects. Practical applications include the cutting of paper reels (Matsumoto et al., 2011), wooden boards (Morabito & Arenales, 2000), glass panels (Durak & Aksu, 2017; Parreño & Alvarez-Valdes, 2021), defective materials (Martin et al., 2020, 2021a), and concrete poles (Lemos et al., 2021). The applications have motivated the development of different solution strategies to tackle problems that vary according to the special requirements of each application field and cutting device. Most of these solution strategies are surveyed and categorized in the works of Dyckhoff (1990), Lodi et al. (2002), Wäscher et al. (2007), Bennell & Oliveira (2008), and Scheithauer (2018), among others. The cutting stock problem and most of its variants are known to be NP-hard (Garey & Johnson, 1979).

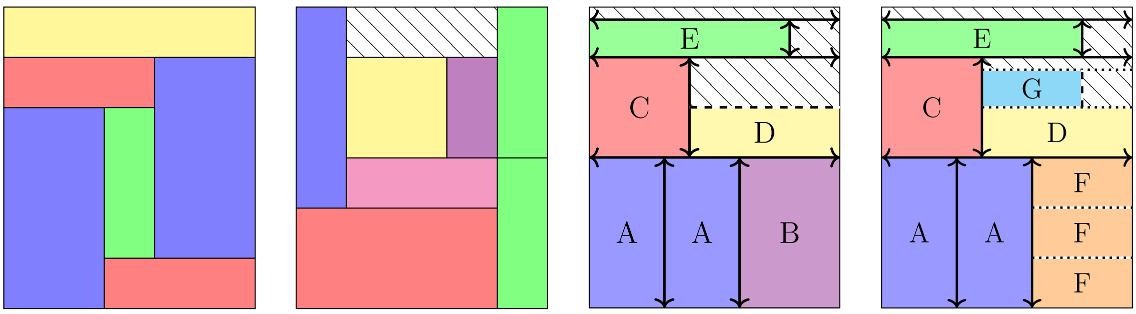
During the cutting sequence of the large objects, all the copies of an item type are stored in a stack, which is usually placed near the cutting machine (Yuen, 1991). The stack of each item type awaits the production of all its corresponding copies to be dispatched to the following manufacturing stage. A stack is called open or in-process during the production of its items, and then it is called closed. In addition to affecting work-in-process levels and delivery times, establishing the cutting sequence gains relevance when the cutting machine holds a limited number of automatic compartments or available space to store the open stacks. For example, Morabito & Belluzzo (2007) addressed a cutting problem that occurs in hardboard companies that employ an automated cutting machine with a limited number of compartments to store the open stacks. In this context, the minimization of open stacks problem (MOSP) is a combinatorial optimization problem that schedules a cutting sequence of the previously determined patterns (i.e., the arrangements of the demanded items into the large objects) while minimizing the maximum number of simultaneously open stacks. This problem appears in several other practical environments apart from cutting operations, as it is also related to various problems pertinent to graph theory (Linhares & Yanasse, 2002; Linhares, 2004). The MOSP is also NP-hard (Yanasse, 1997b).

Although these cutting and sequencing decisions are interdependent in most applications, the 2D-CSP and MOSP are generally treated separately and sequentially in the literature (Martin et al., 2021b). However, this strategy can lead to solutions that are not feasible for the sequencing constraint. Despite their practical relevance, we are not aware of mathematical models that integrate the 2D-CSP and MOSP, that is, the 2D-CSP with a limited number of open stacks. We believe that such exact approaches contribute to finding solutions with satisfactory levels of raw material and open stacks, as the 2D-CSP is highly degenerate in the sense of having several equivalent solutions concerning the use of raw material. The main contributions presented in this paper are: (a) the proposition of an Integer Linear Programming (ILP) modeling approach to integrate the 2D-CSP of 2-stage and 3-stage patterns with a limited number of open stacks; (b) the analysis on the trade-off between raw material solutions and work-in-process levels, that is, the impact on the number of required objects as the allowed number of simultaneously open stacks varies. The proposed ILP formulations have roots in the modeling discussions of Lodi et al. (2004) for the 2D-CSP of 2-stage patterns and Baptiste (2005) for the MOSP. Our models perform best when the number of item types and/or their demands are small. The results show that the the several equivalent solutions of the cutting problem allows obtaining satisfactory waste solutions with a reduced number of open stacks.

The paper is organized as follows. In Section 2, we present a brief review of related literature. In Section 3, we describe the three variants and present an illustrative example to highlight the trade-off between the number of required objects and the number of open stacks. The proposed ILP formulations are presented in Section 4. We start by proposing an ILP formulation for a variant of 2-stage patterns, which is then modified for modeling the other two variants. The computational experiments performed to evaluate the approaches are reported in Section 5. In addition, we describe how to obtain lower and upper bounds for the problems as well as two pre-processing techniques in the Appendix. In Section 6, we present the conclusions of the study and opportunities for future research.

2. Review of the literature

The cutting problems are usually categorized as non-guillotine and guillotine patterns due to the limitations imposed by the cutting device. The guillotine patterns are a sub-class of the non-guillotine patterns, because the guillotine cuts are limited to edge-to-edge cuts. More particularly, an orthogonal guillotine cut on a large rectangle always produces two smaller rectangles. Fig. 1a and 1b depict non-guillotine and guillotine patterns, respectively. Regarding guillotine patterns, practical environments are often interested in increasing the productivity of the cutting machine (e.g., square meters cut per hour) through a sequence of cuts in the same direction. For instance, the pattern of Fig. 1c is 2-stage, since a sequence of three first-stage horizontal cuts generates the strips of items A-A-B, C-D, E, and waste; and then four second-stage vertical cuts split these strips into items. In addition to these cuts,



(a) non-guillotine. (b) guillotine. (c) 2-stage. (d) 3-stage.

Figure 1: Examples of two-dimensional patterns.
Figure 1 Alt Text: Figure 1a. A pattern that does not meet the guillotine constraint; Figure 1b. A pattern that meets the guillotine constraint; Figure 1c. A guillotine pattern in which the items are cut out of the object through two guillotine stages with trimming cuts; Figure 1d. A guillotine pattern in which the items are cut out of the object through three guillotine stages with trimming cuts.

the pattern of Fig. 1d is 3-stage as it needs a third sequence of cuts, depicted as dotted lines, to cut the three items F out of the pile of strip A-A-F-F-F, and items D and G out of strip C-D-G. These d-stage patterns, d $\in \mathbb{N}_+$, are further categorized according to the trimming cuts, that is, the use of an additional guillotine stage to split an item from the waste. The 2-stage pattern of Fig. 1c is non-exact since a trimming cut, depicted as a dashed line, is required to split item D from the waste in strip C-D; conversely, it would be an exact 2-stage pattern if items C and D had the same width. The 3-stage pattern of Fig. 1d is non-exact due to trimming cut that is required to split item G from the waste in strip C-D-G; but it would be an exact 3-stage pattern if items D and G had the same length.

The literature of the 2D-CSP encompasses a broad scope of exact and heuristic approaches relying on ILP formulations, decomposition methods, lower bounds, constructive heuristics, and meta-heuristics to tackle different applications of the problem and its variants (Scheithauer, 2018). Thus, we restrict our attention to papers dealing with ILP formulations for the 2D-CSP of two- and three-stage patterns. Some of these formulations were originally proposed in the context of the two-dimensional bin packing problem, which can be seen as a 2D-CSP of unitary demand for each item type. The compact model of Lodi et al. (2004) for the 2D-CSP of 2-stage patterns assumes that first-stage horizontal cuts generate strips (or levels) with the same length of the object and width equal to the widest item cut out from it, which is the item that initializes the strip. In the model, a set of two-index binary variables represents the fact that each item can initialize a strip in an object. Before mounting the model, the items are sorted by non-increasing width to facilitate the identification of items cut from strips initialized by wider items. Thus, the items are packed into strips, and then the strips are packed into the objects. The model avoids symmetrical solutions, without loss of optimality, by considering that the first item is cut out from the first object, the second item from the first or second objects, the third item from any of the first three objects, and so on. Prior to this work, Lodi & Monaci (2003) proposed models with a similar modeling strategy for the two-dimensional knapsack problem of 2-stage patterns.

Macedo et al. (2010) and Silva et al. (2010) proposed models for the 2D-CSP of 2-stage patterns by extending the formulations of Valério de Carvalho (1999) and Dyckhoff (1981) for the one-dimensional CSP, respectively. These formulations are computationally treatable in instances of the literature despite having a pseudo-polynomial number of variables and constraints. The arc-flow model of Macedo et al. (2010) defines the problem as a set of two one-dimensional problems, where a set of flow problems determines the horizontal strips, and then these strips are packed into the objects with an additional flow problem. The model of Silva et al. (2010) can address the 2D-CSP of 2-stage and 3-stage of patterns depending on the enumeration of the variables. It considers the notion of cuts and residual plates obtained after the cuts. We note that the adaptation of these pseudo-polynomial formulations to approach the sequencing decision of the MOSP is not straightforward because they do not detail from which object an item is cut, which is necessary to determine the number of open stacks. Regarding models for the 2D-CSP of 3-stage patterns, the compact models of Puchinger & Raidl (2007) are based on the modeling strategy of Lodi et al. (2004). The models consider three sets of two-index binary variables to pack items into piles, the piles into strips, and then the strips into objects. Finally, Parreño & Alvarez-Valdes (2021) proposed a model for a variant of the 2D-CSP of 3-stage patterns that addresses large objects with defects in a similar manner, but with a three-index binary variables linking the piles, strips, and objects.

A problem instance of the MOSP is characterized by a binary matrix. In a cutting operations perspective, the rows represent the item types/stacks and the columns the previously determined patterns. The objective is to find a permutation of the columns of this matrix that minimizes the maximum number of simultaneously open stacks. This permutation feature motivates the development of several heuristics and meta-heuristics for the problem (Becceneri et al., 2004; Carvalho & Soma, 2015; Gonçalves et al., 2016; Frinhani et al., 2018). Recently, Martin et al. (2021b) proposed two ILP formulations and a Constraint Programming (CP) model for the MOSP. They compared them with the previous ILP formulations in the literature (Yanasse, 1997b; Baptiste, 2005; Lopes & De Carvalho, 2015). All these five ILP formulations rely on two-index binary variables to establish a sequence of patterns or closed stacks, and then a counting strategy to obtain the maximum number of open stacks. The idea of sequencing the closed stacks comes from the MOSP-graph, that is, an interpretation of the problem as a graph in which the nodes represent the item types/stacks and an edge links two nodes when a pattern produces the corresponding item types (Yanasse, 1997a). The models of Yanasse (1997b), Baptiste (2005), and the second model of Martin et al. (2021b) sequence the patterns, while the model of Lopes & De Carvalho (2015) and the first model of Martin et al. (2021b) sequence the closed stacks. The computational experiments of Martin et al. (2021b) do not report a general winner among the ILP formulations.We note that adapting the MOSP graph-based formulations to take into account pattern generation is not straightforward, since they assume the patterns are already determined and fixed to establish the graph. As for pattern sequencing-based formulations, we chose to consider the constraints from the model of Baptiste (2005), as the other formulations present a weak linear relaxation and/or a large number of big-M constraints.

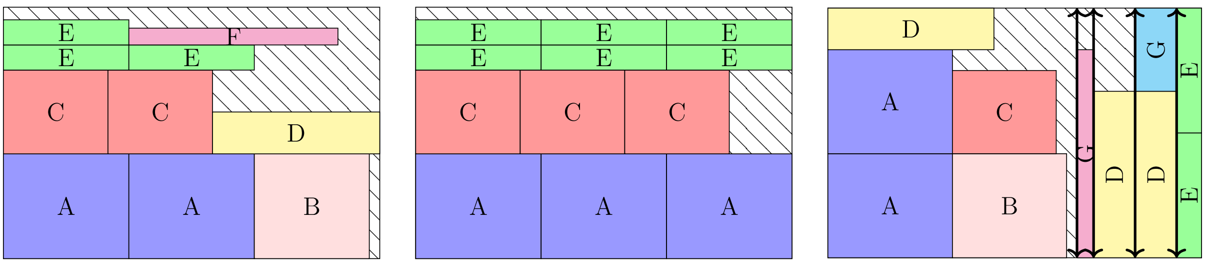
As far as integrated approaches are concerned, Dyson & Gregory (1974) and Madsen (1979, 1988) seem to be the first to address cutting problems with a sequencing constraint. They solved the problem in two stages: the column generation approach of Gilmore & Gomory (1965) first generates the patterns with minimum waste; then, in a second stage, the patterns are sequenced with techniques originally designed for the traveling salesman problem to reduce the number of discontinuities, which are the number of times that the cutting of an item type is re-initiated, or the maximum time a stack remains open. There are a few integrated approaches for the one-dimensional CSP with a limited number of open stacks (Armbruster, 2002; Yanasse & Lamosa, 2007; Pileggi et al., 2005; Arbib et al., 2016). Following the strategies in Pileggi et al. (2005), Pileggi et al. (2007) proposed constructive heuristics for the 2D-CSP with a limited number of open stacks based on column generation and a greedy procedure. Finally, the works of Rinaldi & Franz (2007) and Lucero et al. (2015) proposed, respectively, heuristics and ILP formulations for the strip packing problem with at most two item types cut out of each object.

3. Description of the problems

The Two-dimensional Rectangular Single-Stock Size Cutting Stock Problem (2D-R-SSS-CSP) is a standard problem in the typology of Wäscher et al. (2007) for cutting and packing problems. The 2D-R-SSS-CSP deals with the cutting of a set $I = \{1, \ldots, \overline{M}\}$ of rectangular item types with length l_i, width w_i, and demand d_i, $i \in I$, out of a minimum number of rectangular large objects of length $\overline{L}$ and width $\overline{W}$. In this paper, we address three variants of the 2D-R-SSS-CSP that include the constraints that follow.

- Geometric constraint. Each item must be cut with its edges parallel to its corresponding object's edges and without overlapping the other items cut out of the same object.
- Technological constraint. The cutting machine requires guillotine patterns, i.e., a cutting on a larger rectangle always generates two smaller rectangular limited to two or three guillotine stages.
- Scheduling constraint. The number of simultaneously open stacks during the cutting of the objects must be lower or equal to a scalar K associated with the number of automatic compartments or available space near the cutting machine.

We refer to these variants as integrated problems (IP), given that the scheduling constraint is related to the MOSP. They arise in manufacturing environments that seek minimum waste solutions with limited levels of work-in-process, such as hardboard and flat glass companies. The three IPs differ regarding the type of guillotine pattern employed. The first IP deals with non-exact 2-stage patterns and the second with exact 2-stage patterns that consider only homogeneous strips. A homogeneous strip contains a single item type $i \in I$ with as many copies as possible ($\lfloor \overline{L}/l_i \rfloor$). We denote the first IP by 2stg-IP and the second by 2stg-IP-HS. Both problems assume first-stage horizontal cuts followed by second-stage vertical cuts, like other approaches in the literature (Lodi et al., 2004; Macedo et al., 2010;



(a) 2stg-IP.

(b) 2stg-IP-HS.

(c) 3stg-IP.

Figure 2: Examples of patterns of the 2stg-IP, 2stg-IP-HS, and 3stg-IP.
Figure 2 Alt Text: Figure 2a. A non-exact 2-stage pattern; Figure 2b. An exact 2-stage pattern with homogeneous strips; Figure 2c. A non-exact 3-stage pattern with two sections, where the items are cut out of the second section through 90-degrees rotated exact strips.

Silva et al., 2010). The 2stg-IP allows trimming cuts when necessary, while the 2stg-IP-HS does not. Figs. 2a and 2b depict patterns for the 2stg-IP and 2stg-IP-HS, respectively. In these figures, the exact strips of items A-A-B and E-E in Fig. 2a are feasible for the 2stg-IP; however, they are not feasible for the 2stg-IP-HS. We discuss how to adapt the proposed models to address exact 2-stage patterns in Section 4. Notice that in the 2stg-IP-HS, we opted for 2-stage patterns of homogeneous strips, because they directly facilitate the cutting process and indirectly affect the opening/closing of the stacks.

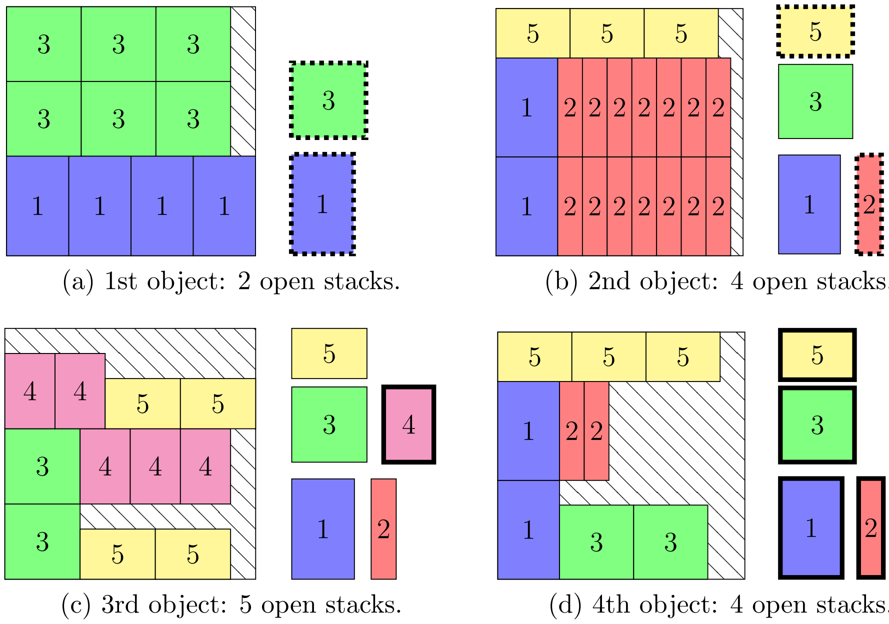
The last IP deals with a specific type of 3-stage patterns, as depicted in Fig. 2c. We denote the problem by 3stg-IP. This type of pattern comes from the use of an automated cutting machine with a limited number of compartments and book rotation (Morabito & Belluzzo, 2007). These machines are present in hardboard industries with high demand and furniture companies with small and medium demand. They assume the production of strips with a set of first-stage vertical cuts, called head cuts, in the right side of the objects, which are depicted as thick arrows in Fig. 2c. Next, the book rotation of the cutting machine allows the rotation of these vertical strips by 90 degrees to obtain items of the same width from the each strip. We note that the length of the right side is up to half of the object's length. Conversely, the left side of the object is not rotated, and the items are obtained from two additional guillotine stages and trimming cuts. One may understand this type of pattern as a 3-stage pattern of two sections in which the first section deals with a non-exact 2-stage pattern and the second with an exact 2-stage pattern of rotated strips.

First, we assume a fixed rotation of items concerning the geometric constraint of the 2stg-IP, 2stg-IP-HS, and 3stg-IP. Thus, two items $l_i = w_j$ and $w_i = l_j$ are considered as distinct. This requirement is essential for cutting anisotropic materials with their growth grains or veins, given that a 90-degree rotation of the items can lead to structural failures in the final products. However, seeking to improve material usage and without incurring failures, one may allow the rotation of a few item types not relevant to the structure of the final product. One example is long and narrow item types ($l_i \gg w_i$, $i \in I$) used on the backs of the furniture pieces. That is the case, for example, of the items cut out of the second section of the patterns of the 3stg-IP. Nevertheless, we emphasize that all of the ILP formulations of Section 4 can meet the rotational requirement by simply adding a new item type i' of length $l'_i = w_i$ and width $w'_i = l_i$, for each $i \in I$, if necessary, aggregating the copies of item types $i, i' \in I$ to fulfill demand d_i.

In Table 1, we present an illustrative example of the 2D-R-SSS-CSP with $\overline{L} = \overline{W} = 100$ and $\overline{M} = 5$ item types. For this example, in Fig. 3, we illustrate an optimal solution for the 2stg-IP assuming $K = 5$ as the maximum number of simultaneously open stacks. In

Table 1 Illustrative example of the 2D-R-SSS-CSP with $\overline{L} = \overline{W} = 100$ and $\overline{M} = 5$ item types.

i	1	2	3	4	5
l_i	25	10	30	20	30
w_i	40	40	30	30	20
d_i	8	16	10	5	10

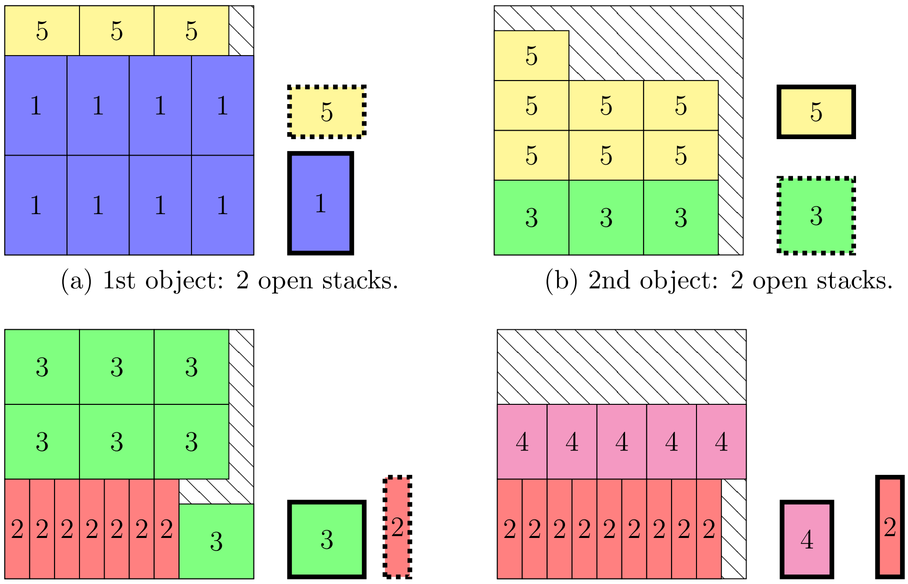


(a) 1st object: 2 open stacks.
(b) 2nd object: 4 open stacks.
(c) 3rd object: 5 open stacks.
(d) 4th object: 4 open stacks.

Figure 3: An optimal solution for the illustrative example assuming up to $K = 5$ open stacks.
Figure 3 Alt Text: Figure 3a. The first pattern produces copies of item types 1 and 3; Figure 3b. The second pattern produces copies of item types 1, 2, and 5; Figure 3c. The third pattern produces copies of item types 3, 4, and 5, where the stack of item type 4 is opened during and closed after this pattern; Figure 3d. The last pattern produces copies of item types 1, 2, 3, and 5, and the stacks of all these four item types are closed after this pattern.

the figure, we show the cutting of four objects with the corresponding open stacks at each stage. For instance, during the cutting of the first object, the stacks of item types 1 and 3 are opened, as illustrated in Fig. 3a with dotted lines. These two stacks are closed only after the processing of the fourth object, as illustrated in Fig. 3d with thick lines. Notice that these integrated problems remain NP-hard when the sequencing constraint is not binding ($K \geq \overline{M}$) due to the cutting problem. Assuming $K = 2$, in Fig. 4, we illustrate an optimal solution for the 2stg-IP. The solutions illustrated in Figs. 3 and 4 require the same number of four objects to produce the items, but the former is not feasible when $K = 2$. From a practical perspective, even in the absence of the present sequencing constraint, the solution of Fig. 4 is preferable to the solution of Fig. 3 as the latter generates less work-in-process during the cutting of the objects.

However, we expect trade-off solutions when the number of required objects tends to increase as the value of K decreases. For example, assuming $K = 1$, any optimal solution requires six objects of homogeneous patterns. A homogeneous pattern contains a single item type $i \in I$ cut from the object with as many copies as possible ($\lfloor \overline{L}/l_i \rfloor \lfloor \overline{W}/w_i \rfloor$). For the illustrative example, to fulfill the demands, each item type requires a single object to fulfill the demands, except item type 3 that needs two objects. Note that any permutation of the sequence of these six objects represents an optimal solution to the problem, assuming the two objects for item type 3 are cut one after the other.



(a) 1st object: 2 open stacks.
(b) 2nd object: 2 open stacks.
(c) 3rd object: 2 open stacks.
(d) 4th object: 2 open stacks.

Figure 4: An optimal solution for the illustrative example assuming up to $K = 2$ open stacks.
Figure 4 Alt Text: Figure 4a. The first pattern produces copies of item types 1 and 5, where the stack of item type 1 is closed after this pattern; Figure 4b. The second pattern produces copies of item types 3 and 5, where the stack of item type 5 is closed after this pattern; Figure 4c. The third pattern produces copies of item types 2 and 3, where the stack of item type 3 is closed after this pattern; Figure 4d. The last pattern produces copies of item types 2 and 4, where the stack of these two item types are closed after this pattern.

4. Mathematical models

In this section, we propose ILP formulations for the 2stg-IP, 2stg-IP-HS, and 3stg-IP. We start by proposing an ILP formulation for the 2stg-IP in Section 4.1, which is then modified to model the 2stg-IP-HS and 3stg-IP in Section 4.2. Recall that a problem instance of these problems is characterized by a tuple $E = (\overline{L}, \overline{W}, \overline{M}, l = (l_1, \ldots, l_{\overline{M}})^\top, w = (w_1, \ldots, w_{\overline{M}})^\top, d = (d_1, \ldots, d_{\overline{M}})^\top, K)$. We assume, without loss of generality, tuple E has positive integers only. The input data of the item types are sorted by non-increasing width, such that, $w_1 \geq w_2 \geq \ldots \geq w_{\overline{M}}$. In addition, we assume any surplus over demand as material waste. We define the following parameters and sets to be used when presenting the models:

$\underline{S}$, $\overline{S}$	lower and upper bounds on the number of required objects, respectively;
$S = \{1, \ldots, \overline{S}\}$	set of objects;
$N_i = \{1, \ldots, \min(\overline{W}/w_i, d_i)\}$	set of horizontal strips initialized by item type $i \in I$;
M^i	upper bound on the number of items of type $i \in I$ that fits in a single object.

The following models rely on the allocation of items to objects. However, unlike other models in the literature, set S is ordinal, so that, it represents the cutting sequence of large objects $s \in S$. Thus, object $s = 1$ is the first to be cut, object $s = 2$ is the second, and so on. This sequence is not optional, as it is associated with the sequencing constraint of the problems.

4.1. An ILP formulation for the 2stg-IP

For the sake of clarity, we next present the formulation for the 2stg-IP in blocks of decisions with their respective explanation. As mentioned before, the modeling approach is rooted in the works of Lodi et al. (2004) and Baptiste (2005). We discuss some modeling choices concerning these works at the end of the section. There are three types of decision variables dealing with the cutting problem in the formulation. The first set concerns the cutting of object $s \in S$, the second refers to the horizontal strips initialized by item type $i \in I$ in object $s \in S$, and the last represents the number of additional items of type $j \in I$ cut out of a strip initialized by item type $i \in I$, $i \leq j$, in object $s \in S$. Thus, we define the following variables:

y^s	binary variable which equals 1, if object $s \in S$ is cut, and 0 otherwise;
z^s_{in}	binary variable which equals 1, if strip $n \in N_i$ of item type $i \in I$ is cut out of object $s \in S$;
x^s_{inj}	integer variable that represents the number of items of type $j \in I$ cut out of strip $n \in N_i$ of item type $i \in I$, $i \leq j$, of object $s \in S$.

An ILP formulation for the 2D-CSP of 2-stage patterns is given by model (1).

$$\textbf{Min} \quad \sum_{s \in S} y^s, \tag{1a}$$

$$\textbf{s.t.} \quad \sum_{s \in S} \sum_{n \in N_i} z^s_{in} + \sum_{s \in S} \sum_{\substack{k \in I, \\ k \leq i}} \sum_{n \in N_k} x^s_{kni} \geq d_i, \qquad i \in I, \tag{1b}$$

$$\sum_{i \in I} \sum_{n \in N_i} w_i z^s_{in} \leq \overline{W} y^s, \qquad s \in S, \tag{1c}$$

$$\sum_{\substack{j \in I, \\ i \leq j}} l_j x^s_{inj} \leq (\overline{L} - l_i) z^s_{in}, \qquad s \in S, i \in I, n \in N_i, \tag{1d}$$

$$y^s \in \{0, 1\}, \qquad s \in S, \tag{1e}$$

$$z^s_{in} \in \{0, 1\}, \qquad s \in S, i \in I, n \in N_i, \tag{1f}$$

$$x^s_{inj} \in \mathbb{Z}_+, \qquad s \in S, i \in I, n \in N_i, j \in I, i \leq j. \tag{1g}$$

The objective function (1a) minimizes the number of objects required to produce the items. Constraints (1b) ensure that the demand of each item type $i \in I$ is fulfilled exactly or in excess. Constraints (1c) ensure that the sum of the strips' widths cut out of object $s \in S$ does not exceed the object's width when $y^s = 1$. Constraints (1d) ensure that the sum of the items' lengths cut out of a strip does not exceed the object's length. Since the input data is sorted by non-increasing width, the summation considers the inequality $i \leq j$

to provide that the item type initializing the strip has the largest width among the items cut from it. Note that $y^s = 0$ enforces the value of variables z^s_{in} to be equal to zero in constraints (1c). Accordingly, in constraints (1d), the value of variables x^s_{inj} is zero when $z^s_{in} = 0$. Constraints (1e) to (1g) define the domain of the decision variables.

There are three types of decision variables dealing with the sequencing decision of the 2stg-IP in the formulation. These variables will also be used in the formulations for the 2stg-IP-HS and 3stg-IP. The first two sets concern the start and end of the processing of the stack of item type $i \in I$ (or shortly, stack $i \in I$). The last set indicates when a stack $i \in I$ is open. Thus, we define the following variables:

b^s_i binary variable which equals 1, if the processing of stack $i \in I$ starts before or at object $s \in S$, and 0 otherwise;
e^s_i binary variable which equals 1, if the processing of stack $i \in I$ ends after or at object $s \in S$, and 0 otherwise;
p^s_i binary variable which equals 1, if stack $i \in I$ is open during the cutting of object $s \in S$, and 0 otherwise.

The following blocks of constraints model the open stacks in the proposed ILP formulation for the 2stg-IP:

$$b^s_i \geq b^{s-1}_i, \qquad s \in S\backslash\{1\}, i \in I, \quad \text{(2a)}$$
$$e^s_i \leq e^{s-1}_i, \qquad s \in S\backslash\{1\}, i \in I, \quad \text{(2b)}$$
$$p^s_i = b^s_i + e^s_i - 1, \qquad s \in S, i \in I, \quad \text{(2c)}$$
$$\sum_{i \in I} p^s_i \leq K y^s, \qquad s \in S, \quad \text{(2d)}$$
$$b^s_i, e^s_i, p^s_i \in \{0, 1\}, \qquad s \in S, i \in I. \quad \text{(2e)}$$

Constraints (2a) and (2b) ensure the definition of variables b^s_i and e^s_i, respectively. Constraints (2c) ensure stack $i \in I$ is open only during the cutting of the objects with $b^s_i = 1$ and $e^s_i = 1$. Alternatively stated, stack $i \in I$ is not open ($p^s_i = 0$), if its processing has either not started ($b^s_i = 0$) or already ended ($e^s_i = 0$). Constraints (2d) limit the number of simultaneously open stacks to scalar K and enforce that no stack is open when $y^s = 0$. Constraints (2e) define the domain of the decision variables. Note that constraints (2c) allow the relaxation of the integrality of one out of these three sets of binary variables. As in Baptiste (2005), we set $p^s_i \in [0, 1]$, $s \in S$, $i \in I$.

In addition to constraints (2d), the previous cutting and sequencing decisions of the 2stg-IP are linked with the following constraints:

$$\sum_{i \in I} \sum_{n \in N_i} z^s_{in} + \sum_{\substack{k \in I, \\ k \leq i}} \sum_{n \in N_k} x^s_{kni} \leq M^i p^s_i, \qquad s \in S, i \in I, \quad (3)$$

where these disjunctive Big-M inequalities ensure that, if an item of type $i \in I$ is cut out of object $s \in S$, then its corresponding stack has to be open. For the 2stg-IP, we define $M^i = \min\{\lfloor \overline{L}/l_i \rfloor \lfloor \overline{W}/w_i \rfloor, d_i\}$, which is a sufficiently large bound for the left-hand-side of the constraints. Alternatively stated, it is the minimum between the number of items of type $i \in I$, assuming a homogeneous pattern and the corresponding demand.

We next impose some expressions to avoid symmetrical solutions:

$$y^s = 1, \qquad s \in \{1, \dots, \underline{S}\}, \tag{4a}$$

$$y^s \geq y^{s+1}, \qquad s \in \{\underline{S}+1, \dots, \overline{S}-1\}, \tag{4b}$$

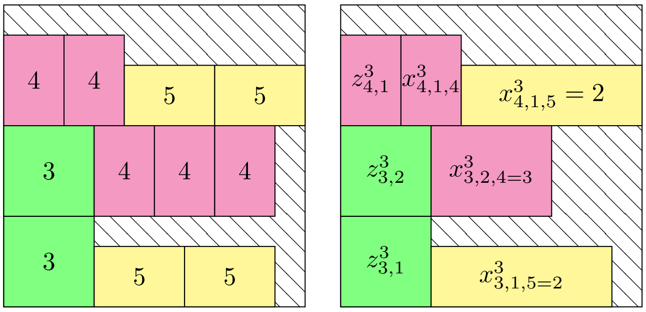
$$\sum_{s \in S} b_k^s \geq \frac{1}{2} \sum_{s \in S} y^s + 1. \tag{4c}$$

Constraints (4a) work as fixing variables given that parameter $\underline{S}$ is a lower bound on the number of objects required. For the remaining objects in set S, valid inequalities (4b) ensure that an object is cut only if its predecessor was also cut. To avoid reverse sequences, valid inequality (4c) enforces that the processing of one stack $k \in I$ starts in the first half of the cutting sequence of objects. The use of this inequality comes from the observation that any cutting sequence and its reverse sequence will always lead to the same maximum number of simultaneously open stacks, differing only from the instants of opening and closing the stacks (Smith & Gent, 2005; Martin et al., 2021b). For instance, the reverse sequence of the solution of Fig. 3 opens the stacks 1, 2, 3, and 5 during the cutting of the first object (the fourth object of Fig. 3), and the maximum number of five open stacks is reached during the processing of the second object (the third object of Fig. 3). In the computational experiments, we arbitrarily chose $k = 1$. Notice that the valid inequalities $z_{in}^s \geq z_{i(n+1)}^s$, $s \in S$, $i \in I$, $n \in N_i \backslash \{\min(\overline{W}/w_i, d_i)\}$ could be included in the model to determine the use of strips initialized by the same item type, without loss of optimality. We chose not to include this block of inequalities because it did not improve performance significantly during the development phase of this study. In fact, it hindered performance by introducing a large number of constraints, depending on the cardinality of the N_i sets.

Having defined all the parameters, variables, and constraints, we present an ILP formulation for the 2stg-IP in model (5). The formulation relies on the cutting problem of expressions (1a) to (1g), the sequencing problem of expressions (2a) to (2e), linking constraints (3), and constraints (4a) to (4c).

$$\begin{aligned} \textbf{Min} \quad & (1a), \\ \textbf{s.t.} \quad & (1b)-(1g), (2a)-(2e), (3), (4a)-(4c). \end{aligned} \tag{5}$$

To exemplify the variables of model (5), we make use of the solution of Fig. 3 with $K = 5$ open stacks. In Fig. 5 we illustrate the cutting of the third object ($s = 3$) with emphasis on the variables having non-zero values. There is one strip initialized by item type $i = 4$, so let us assume $z_{4,1}^3 = 1$. This strip has one additional item of type 4 ($x_{4,1,4}^3 = 1$) and two items



(a) Pattern. (b) $y^3 = 1$.

Figure 5: Illustrative pattern and its corresponding variables of the cutting decision with non-zero values. The value of the illustrated variables in (b) is one when not stated.

Figure 5 Alt Text: Figure 5a. A non-exact 2-stage pattern with three strips, where the first strip is initialized by item type 1 and has two copies of item type 5, the second strip is initialized by item type 3 and has three copies of item type 4, and the last strip is initialized by item type 4 and has one additional copy of item type 4 and two copies of item type 5; Figure 5b. The non-zero values are $z^3_{3,1} = 1$ and $x^3_{3,1,5} = 2$ in the first strip, $z^3_{3,2} = 1$ and $x^3_{3,2,4} = 3$ in the second strip, and $z^3_{4,1} = 1$, $x^3_{4,1,4} = 1$, and $x^3_{4,1,5} = 2$ in the last strip.

of type 5 ($x^3_{4,1,5} = 2$). For the sequencing decisions, in Table 2, we illustrate the values of variables b^s_i, e^s_i, and p^s_i for the opening and closing of stacks $i = 4$ and $i = 5$ of Fig. 3. Let us assume $\overline{S} = 6$, where the last two objects were not necessary in the solution ($y^5 = y^6 = 0$). Recall that stack $i = 5$ is open during the cutting of the second, third, and fourth objects; thus, $p^2_3 = p^3_3 = p^3_4 = 1$. Therefore, $b^2_3 = b^3_3 = \ldots = b^6_3 = 1$ and $e^1_3 = e^2_3 = e^3_3 = e^4_3 = 1$.

Table 2 Representation of the value for the variables of the sequencing decision.

	Variables	Object $s \in S$					
		1	2	3	4	5	6
stack $i = 4$	b^s_4	0	0	1	1	1	1
	e^s_4	1	1	1	0	0	0
	p^s_4	0	0	1	0	0	0
stack $i = 5$	b^s_5	0	1	1	1	1	1
	e^s_5	1	1	1	1	0	0
	p^s_5	0	1	1	1	0	0

In the model of Lodi et al. (2004), their valid inequalities to limit the cutting of the first copy of the first item type (i.e., first item) to object $s = 1$, the second item to objects $s = 1$ or $s = 2$, and so on, eliminate several equivalent solutions, which contributes to the convergence of a branch-and-cut algorithm from a general-purpose ILP solver. But, as they are a virtual sequencing constraint, they can lead to loss of optimality if used in conjunction with the sequencing decisions of the MOSP. Alternatively stated, if both were used together, then the opening and closing of the stacks would inadvertently be limited to certain objects. Therefore, we do not consider this type of valid inequalities. Instead, we chose to aggregate the cutting decisions of similar items into item types and index them in sets S and I to allow the link with the sequencing decisions. As for the model of Baptiste (2005), we adapt the block of constraints (2d) to consider that an object may not be cut ($y^s = 0$), which differs from the original model that sequenced all the objects.

4.2. ILP formulations for the 2stg-IP-HS and 3stg-IP

The productivity of the cutting machine varies with the cutting of objects from simpler to more complex patterns regarding the number of guillotine stages. For example, exact 2-stage patterns may be preferred over non-exact 2-stage patterns in periods of high demand despite the risk of a greater amount of waste. Conversely, material usage can be decreased with 3-stage patterns in periods of low demand. In this context, we can adapt model (5) to address the 2stg-IP-HS by simply fixing variables x^s_{inj}, $s \in S$, $i \in I$, $n \in N_i$, $i < j$, to the value of zero. Model (5) can also be adapted to address the 2D-CSP of exact 2-stage patterns or general 3-stage patterns. To the former, we fix variables x^s_{inj}, $s \in S$, $i \in I$, $n \in N_i$, $i < j$, $w_i \neq w_j$, to the value of zero. To the latter, the notion of sections must be included in the model. For instance, assuming $\overline{L} \gg \overline{W}$, first-stage vertical cuts split the objects into sections, then second-stage horizontal cuts generate strips out of the sections, and finally the items are cut out of the strips with third-stage vertical cuts. However, in what follows, we propose ILP formulations for the 2stg-IP-HS and 3stg-IP that directly address the corresponding types of patterns. We chose a similar notation to allow the three formulations to be easily compared in terms of their variables and constraints for the cutting decisions. In addition to material usage, we analyze the performance of the formulations in Section 5 with regard to the relative impact of these cutting and sequencing decisions on the number of variables and constraints.

The formulation for the 2stg-IP-HS has two types of decision variables for dealing with the cutting problem. The first set concerns the cutting of object $s \in S$. The second refers to the number of homogeneous strips of item type $i \in I$ cut out of object $s \in S$. Thus, we define the following variables:

- y^s binary variable which equals 1, if object $s \in S$ is cut, and 0 otherwise;
- z^s_i integer variable that represents the number of homogeneous strips of item type $i \in I$ cut out of object $s \in S$.

An ILP formulation for the 2stg-IP-HS is given by model (6).

$$\textbf{Min} \quad \sum_{s \in S} y^s, \tag{6a}$$

$$\textbf{s.t.} \quad (2a)-(2e), (4a)-(4c),$$

$$\sum_{s \in S} \lfloor \overline{L}/l_i \rfloor z^s_i \geq d_i, \qquad i \in I, \tag{6b}$$

$$\sum_{i \in I} w_i z^s_i \leq \overline{W} y^s, \qquad s \in S, \tag{6c}$$

$$\lfloor \overline{L}/l_i \rfloor z_i^s \leq M^i p_i^s, \qquad s \in S, i \in I, \quad (6d)$$
$$y^s \in \{0,1\}, \qquad s \in S, \quad (6e)$$
$$z_i^s \in \mathbb{Z}_+, z_i^s \leq \lfloor \overline{W}/w_i \rfloor, \qquad s \in S, i \in I. \quad (6f)$$

The objective function (6a) minimizes the number of objects required to produce the items, likewise the model of the previous section. The formulation also relies on the sequencing problem of constraints (2a) to (2e) and expressions (4a) to (4c). Constraints (6b) ensure that the demand of each item type $i \in I$ is fulfilled exactly or in excess. They assume that each homogeneous strip of an item type $i \in I$ produces $\lfloor \overline{L}/l_i \rfloor$ copies. Constraints (6c) ensure that the sum of the strips' width cut out of object $s \in S$ does not exceed the object's width when $y^s = 1$. Note that $y^s = 0$ enforces the value of variables z_i^s to be equal to zero. In addition to constraints (2d), constraints (6d) are responsible for linking the cutting and sequencing decisions of the 2stg-IP-HS, where these disjunctive Big-M inequalities ensure that if at least one homogeneous strip of item type $i \in I$ is cut out of object $s \in S$ then its corresponding stack has to be open. Concerning parameter M^i for the 2stg-IP-HS, We make a small adjustment from the definition of the previous model in regard to parameter M^i for the 2stg-IP-HS, such that, $M^i = \{\lfloor \overline{L}/l_i \rfloor \lfloor \overline{W}/w_i \rfloor, \lceil d_i / \lfloor \overline{L}/l_i \rfloor \rceil \lfloor \overline{L}/l_i \rfloor\}$, where the second term ensure the necessary number of homogeneous strips $i \in I$ if $d_i < \lfloor \overline{L}/l_i \rfloor \lfloor \overline{W}/w_i \rfloor$. Constraints (6e) and (6f) define the domain of the decision variables for the cutting decision. Notice that the adaptation of model (6) to address exact 2-stage patterns of heterogeneous strips would require an additional set of variables to perform a similar role of variables x_{inj}^s in model (5).

Before introducing the formulation for the 3stg-IP, we define the following sets:

$E = \{a, b\}$	set of sections;
$I' = \{i \in I \mid w_i < l_i \leq \overline{W}\}$	set of item types allowed to be cut out of a section b;
$N_i^a = \{1, \dots, \min(\overline{W}/w_i, d_i)\}$	set of horizontal strips initialized by item type $i \in I$;
$N_i^b = \{1, \dots, \min(\overline{L}/(2w_i), d_i)\}$	set of vertical strips initialized by a rotated item of type $i \in I'$.

We denote the left and right sides of the patterns of the 3stg-IP as sections a and b, respectively. Thus, section a deals with a non-exact 2-stage pattern and section b with an exact 2-stage pattern of rotated strips. Following the discussion of Section 3, we assume that only a few item types can be cut out of rotated strips in a section b, which is limited up to half of the object's length. We refer to this subset as belonging to set I' to easily allow the use of benchmark instances from the literature in the next section of this paper. The previous sets N_i, $i \in I$, are now redefined in sets N_i^e, $e \in E$. The formulation has four sets of decision variables for dealing with the cutting problem. Like before, the first set concerns the cutting of object $s \in S$. The second set refers to the length of section $e \in E$ of object $s \in S$. The last two sets indicate when an item type $i \in I$ initializes strip $n \in N_i^e$ in section $e \in E$ of object $s \in S$, or the number of additional items of type $j \in I$ cut out from it. Thus, we define the following variables:

y^s binary variable which equals 1, if object $s \in S$ is cut, and 0 otherwise;
l^{se} length of section $e \in E$ of object $s \in S$;
z^{se}_{in} binary variable which equals 1, if a strip $n \in N^e_i$ of item type $i \in I$ is cut out of section $e \in E$ of object $s \in S$;
x^{se}_{inj} integer variable that represents the number of items of type $j \in I$ cut out of strip $n \in N^e_i$ of item type $i \in I$, $i \leq j$, of section $e \in E$ of object $s \in S$; $w_i = w_j$ holds if $e = b$.

An ILP formulation for the 3stg-IP is given by model (7).

$$\textbf{Min} \quad \sum_{s \in S} y^s, \tag{7a}$$

$$\textbf{s.t.} \quad (2a) - (2e), (4a) - (4c),$$

$$\sum_{s \in S} \sum_{e \in E} \sum_{n \in N^e_i} z^{se}_{in} + \sum_{s \in S} \sum_{e \in E} \sum_{\substack{k \in I \\ k \leq i}} \sum_{n \in N^e_k} x^{se}_{kni} \geq d_i, \qquad i \in I, \tag{7b}$$

$$\sum_{e \in E} l^{se} \leq \overline{L} y^s, \qquad s \in S, \tag{7c}$$

$$\sum_{i \in I} \sum_{n \in N^a_i} w_i z^{sa}_{in} \leq \overline{W} y^s, \qquad s \in S, \tag{7d}$$

$$\sum_{\substack{j \in I \\ i \leq j}} l_j x^{sa}_{inj} \leq (\overline{L} - l_i) z^{sa}_{in}, \qquad s \in S, i \in I, n \in N^a_i, \tag{7e}$$

$$l_i z^{sa}_{in} + \sum_{\substack{j \in I \\ i \leq j}} l_j x^{sa}_{inj} \leq l^{sa}, \qquad s \in S, i \in I, n \in N^a_i, \tag{7f}$$

$$\sum_{i \in I'} \sum_{n \in N^b_i} w_i z^{sb}_{in} = l^{sb}, \qquad s \in S, \tag{7g}$$

$$\sum_{\substack{j \in I' \\ i \leq j}} l_j x^{sb}_{inj} \leq (\overline{W} - l_i) z^{sb}_{in}, \qquad s \in S, i \in I', n \in N^b_i, \tag{7h}$$

$$\sum_{e \in E} \sum_{n \in N^e_i} z^{se}_{in} + \sum_{e \in E} \sum_{\substack{k \in I \\ k \leq i}} \sum_{n \in N^e_k} x^{se}_{kni} \leq M^i p^s_i, \qquad s \in S, i \in I, \tag{7i}$$

$$y^s \in \{0, 1\}, \qquad s \in S, \tag{7j}$$

$$z^{se}_{in} \in \{0, 1\}, \qquad s \in S, e \in E, i \in I, n \in N^e_i, \tag{7k}$$

$$x^{sa}_{inj} \in \mathbb{Z}_+, \qquad s \in S, i \in I, n \in N^a_i, j \in I, i \leq j, \tag{7l}$$

$$x^{sb}_{inj} \in \mathbb{Z}_+, \qquad s \in S, i \in I', n \in N^b_i, j \in I', i \leq j, w_i = w_j, \tag{7m}$$

$$0 \leq l^{sa} \leq \overline{L}, \qquad s \in S, \tag{7n}$$

$$0 \leq l^{sb} \leq \lfloor \overline{L}/2 \rfloor, \qquad s \in S. \tag{7o}$$

The objective function (7a) minimizes the number of objects required to produce the items, as with the previous models. The formulation also relies on the the sequencing problem of constraints (2a) to (2e) and expressions (4a) to (4c). Constraints (7b) ensure that the demand of each item type $i \in I$ is fulfilled exactly or in excess. Constraints (7c) ensure that the sum of the length of the sections does not exceed the object's length when $y^s = 1$; conversely, their value is set to zero when $y^s = 0$.

Note that constraints (7d), (7e), and (7f) model the left side of the objects, as denoted by index a, and constraints (7g) and (7h) model the right side of the objects with rotated items, as denoted by index b. Constraints (7d) ensure that the sum of the strips' width cut out of object $s \in S$ does not exceed the object's width when $y^s = 1$; otherwise, the left hand side is set to zero. Constraints (7e) set the value of zero for variables x^{sa}_{inj} when $z^{sa}_{in} = 0$. Constraints (7f) establish the value of variables l^{sa} according to the length of their corresponding strips; here, the signal $\leq$ is necessary because the length of their corresponding strips is not equal. In fact, many of them are set to zero when $z^{sa}_{in} = 0$. Conversely, constraints (7g) establish the value of variables l^{sb} according to the length of their corresponding strips, that are rotated in 90-degrees, relying only on variables z^{sb}_{in} with a sign of equality. Constraints (7h) ensure that the sum of the items' length cut out of a rotated strip does not exceed the object's width.

Constraints (7i) link the cutting and sequencing decisions of the 3stg-IP. Similar to the previous models, these disjunctive Big-M inequalities ensure that, if an item of type $i \in I$ is cut out of object $s \in S$, then its corresponding stack has to be open. For this model, we define $M^i = \{\lfloor \overline{L}\,\overline{W}/(l_i w_i) \rfloor, d_i\}$, thus making it the relaxed version of the previous definitions of the parameter, given that the items cut from a section b are obtained in rotated strips. Constraints (7j) to (7o) define the domain of the decision variables for the cutting decision.

5. Computational experiments

In this section, we report the computational experiments performed to evaluate the proposed formulations. The purpose is to verify their performance in terms of solution quality and processing time to obtain trade-off solutions when varying the number K of the maximum number of simultaneously open stacks. This section is divided into two parts, and the benchmark instances used in the experiments are detailed at the beginning of these sections. In what follows, we refer to model (5) for the 2stg-IP as Model-2stg-IP. Similarly, model (6) for the 2stg-IP-HS and model (7) for the 3stg-IP are referred to as Model-2stg-IP-HS and Model-3stg-IP, respectively. We compare our models with each other, as we are not aware of any other model in the literature for the 2D-CSP with a limited number of open stacks. Model-2stg-IP, Model-2stg-IP-HS, and Model-3stg-IP were coded in C++ using Gurobi v.9.1.1 as the general-purpose ILP solver. In the Appendix, we describe how we obtained the lower $\underline{S}$ and upper $\overline{S}$ bounds required in the models. We also present two simple pre-processing operations to reduce the size of the models. All experiments were carried out on a PC with Intel Xeon E5-2680v2 (2.8 GHz), 10 threads, 16 GB RAM, under a CentOS Linux 7.2.1511 Operating System. Each run of the solver was limited to 3,600 seconds; we use letters “tl” in the tables to indicate when this time limit was reached for a class of instances.

5.1. Results for the set of instances #A

The set of instances #A consists of the twelve gcut1-12 instances (Beasley, 1985). The size $\overline{L} \times \overline{W}$ of the large objects is 250×250 for gcut1-4 instances, 500×500 for gcut5-8 instances, and 1000×1000 for gcut9-12 instances. The number of item types $\overline{M}$ is 10, 20, 30, or 50. The length l_i and width w_i of item type $i \in I$ were sampled in the intervals $[\overline{L}/4, 3\overline{L}/4]$ and $[\overline{W}/4, 3\overline{W}/4]$, respectively. The demand d_i of item type $i \in I$ is given by $\lfloor \overline{L}/l_i \rfloor \lfloor \overline{W}/w_i \rfloor$ and the average number of demanded items per instance is 80.83. These instances are available in online libraries, such as ESICUP[1] and OR-Library[2], and they were initially proposed for the Single Large Object Placement Problem (Wäscher et al., 2007). For the experiments, we considered two cases concerning the sequencing decisions on the number K of the maximum number of simultaneously open stacks. For the first case, we assume parameter K to vary from 1 to 5 to obtain solutions with a reduced number of open stacks, as a production might prefer for practical reasons. For example, Morabito & Belluzzo (2007) described a cutting machine with five compartments to store the open stacks. For the other case, we considered no sequencing decisions, which is denoted by symbol κ. That is, we experimented with the corresponding variant of each model concerning the cutting decisions only, where variables b_i^s, e_i^s, and p_i^s, $s \in S$, $i \in I$, and the corresponding constraints were dropped out of the models. Our aim was to analyze the relative burden between the cutting and sequencing decisions using the proposed models. Thus, each of the gcut1-12 instances provided six problem instances in a total of 72 ($= 12 \times 6$) instances.

We report the results for the set of instances #A in Table 3. For each model, they are aggregated according to the number of item types $\overline{M}$ and the value of parameter K. Each entry of the table is an average over three instances, except those in the last row and columns OPT. For each model and group of instances, we report the values of the lower bound ($\underline{S}$), objective function (OFV), linear relaxation (LR), optimality gap in percentage (gap[%]), processing time in seconds (time[s]), and number of solutions with optimality proven by the solver (OPT). We present average values in the last row of the table, except in column OPT, which is the summation of the entries. Beyond the integrality of the variables, expressions (4a) for fixing variables $y^s = 1$, $s \in \{1, \dots, \underline{S}\}$ were also relaxed to obtain the values of the linear relaxation. The calculation of the average processing time includes the case when the time limit was reached.

The results in Table 3 show that, for instances in set #A, the average optimality gap of the solver with Model-2stg-IP-HS, Model-2stg-IP, and Model-3stg-IP were 0.77%, 4.08%, and 11.04%, with the average processing time of 674.87 s, 1,334.31 s and 2,342.03 s, respectively. The solver was able to find an optimal solution and prove its optimality in 81.94% of the instances (59 out of 72) with Model-2stg-IP-HS, in 66.66% (48 instances) with Model-2stg-IP, and 36.11% (26 instances) with Model-3stg-IP. Despite a reasonable number of

[1] `https://www.euro-online.org/websites/esicup/data-sets/`
[2] `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`

proven optimal solutions for a class of problems that integrate two NP-Hard problems, these results clearly show that computational results get worse in terms of solution quality and processing time as the patterns become more general and complex. This explains why the average value of the objective function for the Model-2stg-IP-HS, Model-2stg-IP, and Model-3stg-IP were 22.99, 20.86, and 20.90, respectively. In other words, Model-2stg-IP's solutions were equal or better than Model-2stg-IP-HS's solutions in all groups of instances in terms of the number of required objects, but Model-3stg-IP's solutions surpassed Model-2stg-IP's solutions only when $\overline{M} = 10, 20$ item types. As expected, the average gap and average processing time tend to increase as the number of item types $\overline{M}$ also increases. Notice that the average processing time is short when $K = 1$ or without sequencing decisions (κ). However, it increases significantly when the values of K and/or $\overline{M}$ also increase. Alternatively stated, the integration of the cutting and sequencing decisions did not make it easier to solve the models in the context of a branch-and-cut of a general-purpose ILP solver.

Table 3 Results for the set of instances #A.

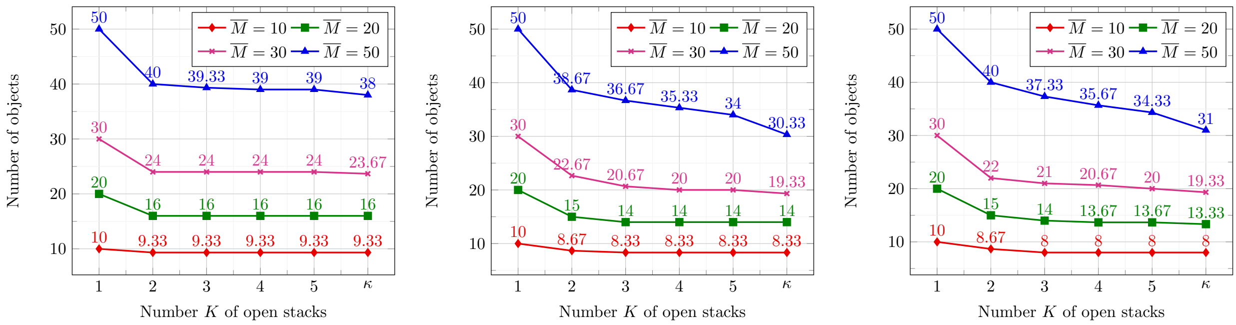
$\overline{M}$	K	Model-2stg-IP-HS						Model-2stg-IP						Model-3stg-IP					
		$\underline{S}$	OFV	LR	gap[%]	time[s]	OPT	$\underline{S}$	OFV	LR	gap[%]	time[s]	OPT	$\underline{S}$	OFV	LR	gap[%]	time[s]	OPT
10	1	10.00	10.00	10.00	0.00	<0.01	3	10.00	10.00	10.00	0.00	<0.01	3	10.00	10.00	10.00	0.00	<0.01	3
	2	9.33	9.33	8.63	0.00	0.01	3	8.67	8.67	7.11	0.00	56.00	3	8.33	8.67	6.01	3.70	2,141.25	2
	3	9.33	9.33	8.63	0.00	0.01	3	8.33	8.33	7.11	0.00	0.51	3	8.00	8.00	5.89	0.00	27.68	3
	4	9.33	9.33	8.63	0.00	0.01	3	8.33	8.33	7.11	0.00	0.28	3	8.00	8.00	5.89	0.00	21.21	3
	5	9.33	9.33	8.63	0.00	0.01	3	8.33	8.33	7.11	0.00	0.12	3	8.00	8.00	5.89	0.00	16.68	3
	κ	9.33	9.33	8.63	0.00	<0.01	3	8.33	8.33	7.11	0.00	0.03	3	8.00	8.00	5.89	0.00	1.66	3
20	1	20.00	20.00	20.00	0.00	<0.01	3	20.00	20.00	20.00	0.00	0.01	3	20.00	20.00	20.00	0.00	0.01	3
	2	16.00	16.00	15.40	0.00	12.84	3	14.00	15.00	12.38	6.69	tl	0	11.67	15.00	10.00	22.22	tl	0
	3	16.00	16.00	15.40	0.00	4.69	3	14.00	14.00	12.38	0.00	823.09	3	11.67	14.00	7.14	16.67	tl	0
	4	16.00	16.00	15.40	0.00	2.39	3	14.00	14.00	12.38	0.00	24.07	3	11.67	13.67	7.14	14.65	tl	0
	5	16.00	16.00	15.40	0.00	1.84	3	14.00	14.00	12.38	0.00	11.49	3	11.67	13.67	7.14	14.65	tl	0
	κ	16.00	16.00	15.40	0.00	0.03	3	14.00	14.00	12.38	0.00	0.42	3	12.00	13.33	7.14	9.89	tl	0
30	1	30.00	30.00	30.00	0.00	<0.01	3	30.00	30.00	30.00	0.00	0.02	3	30.00	30.00	30.00	0.00	0.03	3
	2	23.67	24.00	22.66	1.33	1,534.99	2	19.33	22.67	17.06	14.62	tl	0	17.33	22.00	15.00	21.21	tl	0
	3	23.67	24.00	22.66	1.33	1,252.37	2	19.33	20.67	17.06	6.35	2,666.93	1	17.00	21.00	11.95	19.05	tl	0
	4	23.67	24.00	22.66	1.33	1,204.69	2	19.33	20.00	17.06	3.33	2,539.22	1	17.00	20.67	11.91	17.70	tl	0
	5	23.67	24.00	22.66	1.33	1,222.81	2	19.33	20.00	17.06	3.33	2,445.88	1	17.00	20.00	11.91	15.00	tl	0
	κ	23.67	23.67	22.66	0.00	0.32	3	19.33	19.33	17.06	0.00	11.03	3	17.33	19.33	11.91	10.35	tl	0
50	1	50.00	50.00	50.00	0.00	0.01	3	50.00	50.00	50.00	0.00	0.13	3	50.00	50.00	50.00	0.00	0.14	3
	2	38.00	40.00	36.92	4.92	tl	0	30.33	38.67	28.25	21.52	tl	0	28.33	40.00	25.00	29.09	tl	0
	3	38.00	39.33	36.92	3.29	2,509.22	1	30.33	36.67	28.25	17.27	tl	0	28.33	37.33	16.84	24.09	tl	0
	4	38.00	39.00	36.92	2.50	2,418.43	1	30.33	35.33	28.25	14.09	tl	0	28.33	35.67	16.45	20.56	tl	0
	5	38.00	39.00	36.92	2.50	2,428.22	1	30.33	34.00	28.25	10.76	tl	0	28.33	34.33	16.45	17.48	tl	0
	κ	38.00	38.00	36.92	0.00	3.99	3	30.33	30.33	28.25	0.00	1,844.32	3	28.33	31.00	16.45	8.57	tl	0
Avg./Sum		*22.71*	*22.99*	*22.00*	*0.77*	*674.87*	*59*	*19.60*	*20.86*	*18.08*	*4.08*	*1,334.31*	*48*	*18.18*	*20.90*	*13.83*	*11.04*	*2,342.03*	*26*

Notes:

(1) parameter $K = 1, 2, 3, 4, 5$ is the maximum number of simultaneously open stacks, and symbol κ denotes that the sequencing decisions were dropped from the models.

(2) each entry in this table is an average over three instances, except those in the last row and columns OPT.

(3) the last row presents averages in columns $\underline{S}$, OFV, LR, gap[%] and time[s], and summations in columns OPT.



(a) Model-2stg-IP-HS. (b) Model-2stg-IP. (c) Model-3stg-IP.

Figure 6: Pareto curves for the Model-2stg-IP-HS, Model-2stg-IP, and Model-3stg-IP for the number of required objects and number K of maximum number of simultaneously open stacks for the set of instances #A.

Figure 6 Alt Text: Figure 6a. The number of required objects has a sharp drop from $K = 1$ to 2 open stacks and is practically constant from K=2 onwards; Figure 6b. The number of required objects has a sharp drop from $K = 1$ to 2 open stacks and a slight variation from K = 2 onwards, unless for $\overline{M} = 50$ item types; Figure 6c. The number of required objects has a sharp drop from $K = 1$ to 2 open stacks and a slight variation from K = 2 onwards, unless for $\overline{M} = 50$ item types.

We illustrate the trade-off between the number of required objects (columns OFV) and parameter K for each model in Fig. 6. The Pareto curves of the Model-2stg-IP-HS are optimal when $\overline{M} = 10, 20$ and, for Model-2stg-IP, when $\overline{M} = 10$. The analysis of all these Pareto curves shows a sharp drop in the number of required objects from $K = 1$ to 2 open stacks. Next, the variation in the reduction of this number decreases significantly as the value of parameter K increases, especially for smaller values in the number of item types $\overline{M}$ or simpler patterns like those of Model-2stg-IP-HS. We highlight that many of the solutions obtained when neglecting the sequencing decisions (κ) are not even feasible for the scenarios $K = 1, 2, 3, 4, 5$. For instance, assuming the cutting sequence given by the ordinal set S, the average number of simultaneously open stacks was 11.37 for scenarios κ with 75.00 % of the solutions having more than 5 simultaneously open stacks. Note that the number of simultaneously open stacks tends to increase as the number of item types per object also increases. In particular, the average number of item types per object was 1.75 and 2.46 for scenarios $K = 1, 2, 3, 4, 5$ and κ, respectively. This is an indication of why solving the cutting and sequencing problems in a separate and sequential manner can lead to infeasible solutions.

In Table 4, we report the average number of variables (var) and constraints (cons) for each model. We aggregate these values into when the sequencing constraints are binding ($K = 1, 2, 3, 4, 5$) and when they are not (κ). For the three models, the number of variables and constraints increases quickly as the number of item types $\overline{M}$ increases. In the Model-2stg-IP-HS with $\overline{M} = 10$, the variables and constraints for the cutting decisions represent approximately 25.00% (i.e., 100.67 out of 400.53 variables) and 5.00% (i.e., 20.00 out of 411.00 constraints) of the total number of variables and constraints of the model, respectively. On average, these ratios of the variables and constraints are, respectively, 25.11% and 2.58%

for Model-2stg-IP-HS, 85.23% and 31.52% for Model-2stg-IP, and 85.61% and 51.17% for Model-3stg-IP. This shows that the blocks of constraints (2a)-(2e) and the corresponding linking constraints for the sequencing decisions make up a preponderant of the total variables in the proposed models.

Table 4 Number of variables and constraints of the models for the set of instances #A.

		Model-2stg-IP-HS		Model-2stg-IP		Model-3stg-IP	
		var	cons	var	cons	var	cons
$\overline{M}$=10	$K = 1,2,3,4,5$	400.53	411.00	1,311.33	584.87	1,352.13	829.00
	κ	100.67	20.00	1,011.67	194.00	1,052.67	438.33
$\overline{M}$=20	$K = 1,2,3,4,5$	1,603.20	1,623.40	7,718.13	2,331.67	7,986.67	3,306.87
	κ	404.00	43.00	6,519.33	751.67	6,788.33	1,727.33
$\overline{M}$=30	$K = 1,2,3,4,5$	3,605.07	3,635.27	22,638.53	5,128.73	23,170.40	7,150.60
	κ	906.33	65.33	19,940.67	1,559.67	20,473.00	3,582.00
$\overline{M}$=50	$K = 1,2,3,4,5$	10,009.60	10,059.80	98,232.40	14,399.27	99,834.00	20,200.87
	κ	2,512.00	111.00	90,736.33	4,452.00	92,338.33	10,254.00

5.2. Results for the set of instances #B

The set of instances #B consists of 30 instances proposed by Yanasse & Morabito (2006, 2008). The size $\overline{L} \times \overline{W}$ of the large objects is 100×100. The number of item types $\overline{M}$ is 10, 20, or 50. The length l_i and width w_i of item type $i \in I$ were sampled in the intervals $[0.1\overline{L}, 0.5\overline{L}]$ and $[0.1\overline{W}, 0.5\overline{W}]$, respectively. The demand d_i of item type $i \in I$ were uniformly sampled in the interval $[1, \lfloor \overline{L}/l_i \rfloor \lfloor \overline{W}/w_i \rfloor]$. All these values were sampled and then rounded. The average number of items per instance is 222.70; thus, the items in set #B are smaller than the items in set #A. The instances are available upon request to the authors. There are 10 problem instances for each class of item types, in a total of 30 ($= 10 \times 3$) instances. Using the structure of the previous section, we report the results for the set of instances #B in Tables 5 and 6.

Table 5 Results for the set of instances #B.

		Model-2stg-IP-HS						Model-2stg-IP						Model-3stg-IP					
$\overline{M}$	K	$\underline{S}$	OFV	LR	gap[%]	time[s]	OPT	$\underline{S}$	OFV	LR	gap[%]	time[s]	OPT	$\underline{S}$	OFV	LR	gap[%]	time[s]	OPT
10	1	10.70	10.70	9.19	0.00	<0.01	10	10.70	10.70	10.12	0.00	<0.01	10	10.70	10.70	10.09	0.00	<0.01	10
	2	7.10	7.10	5.37	0.00	2.16	10	6.20	6.20	5.19	0.00	37.12	10	6.00	6.10	5.04	1.25	378.13	9
	3	6.50	6.50	5.26	0.00	0.07	10	5.30	5.60	4.45	4.76	1,139.75	7	5.10	5.40	3.36	4.76	1,131.77	7
	4	6.50	6.50	5.26	0.00	0.04	10	5.20	5.40	4.39	3.10	726.28	8	5.00	5.30	2.72	5.10	1,156.67	7
	5	6.50	6.50	5.26	0.00	0.02	10	5.20	5.30	4.39	1.67	426.03	9	4.90	5.10	2.64	3.43	799.78	8
	κ	6.50	6.50	5.26	0.00	<0.01	10	5.20	5.20	4.39	0.00	152.94	10	5.00	5.10	2.61	1.67	434.83	9
20	1	21.10	21.10	18.84	0.00	<0.01	10	21.10	21.10	20.23	0.00	0.01	10	21.10	21.10	20.19	0.00	0.02	10
	2	13.30	13.70	11.52	2.94	1,441.89	6	10.90	12.70	10.13	13.95	tl	0	10.90	12.50	10.10	12.34	3,252.91	1
	3	13.30	13.30	11.52	0.00	1.09	10	10.60	11.80	9.55	9.97	3,317.09	1	10.10	12.00	6.73	15.69	tl	0
	4	13.30	13.30	11.52	0.00	0.70	10	10.60	11.50	9.55	7.85	3,275.58	1	10.10	11.50	5.45	12.17	tl	0
	5	13.30	13.30	11.52	0.00	0.20	10	10.60	11.40	9.55	6.94	2,980.40	2	10.10	11.40	5.30	11.35	tl	0
	κ	13.30	13.30	11.52	0.00	0.02	10	10.60	10.90	9.55	2.50	1,304.15	7	10.10	11.00	5.30	7.97	3,313.97	1
50	1	55.60	55.60	47.51	0.00	0.01	10	55.60	55.60	51.22	0.00	0.23	10	55.60	55.60	51.09	0.00	0.35	10
	2	33.40	35.60	29.50	6.19	tl	0	28.10	36.90	25.75	23.22	tl	0	28.10	41.00	25.54	30.84	tl	0
	3	33.40	34.80	29.50	4.01	tl	0	25.30	32.40	24.22	21.89	tl	0	24.60	36.50	17.03	31.53	tl	0
	4	33.40	34.40	29.50	2.92	tl	0	25.40	30.90	24.22	17.77	tl	0	24.60	34.80	13.44	27.84	tl	0
	5	33.40	34.00	29.50	1.76	2,617.29	4	25.40	30.30	24.22	16.17	tl	0	24.60	33.30	13.04	25.04	tl	0
	κ	33.40	33.50	29.50	0.29	680.35	9	25.30	27.20	24.22	6.95	tl	0	24.60	28.30	13.04	13.05	tl	0
Avg./Sum		*19.67*	*19.98*	*17.06*	*1.01*	*863.55*	*139*	*16.52*	*18.39*	*15.30*	*7.60*	*1,942.20*	*85*	*16.18*	*19.26*	*11.82*	*11.33*	*2,181.58*	*72*

Notes:

(1) parameter $K = 1, 2, 3, 4, 5$ is the maximum number of simultaneously open stacks, and symbol κ denotes that the sequencing decisions were dropped from the models.

(2) each entry in this table is an average over ten instances, except those in the last row and columns OPT.

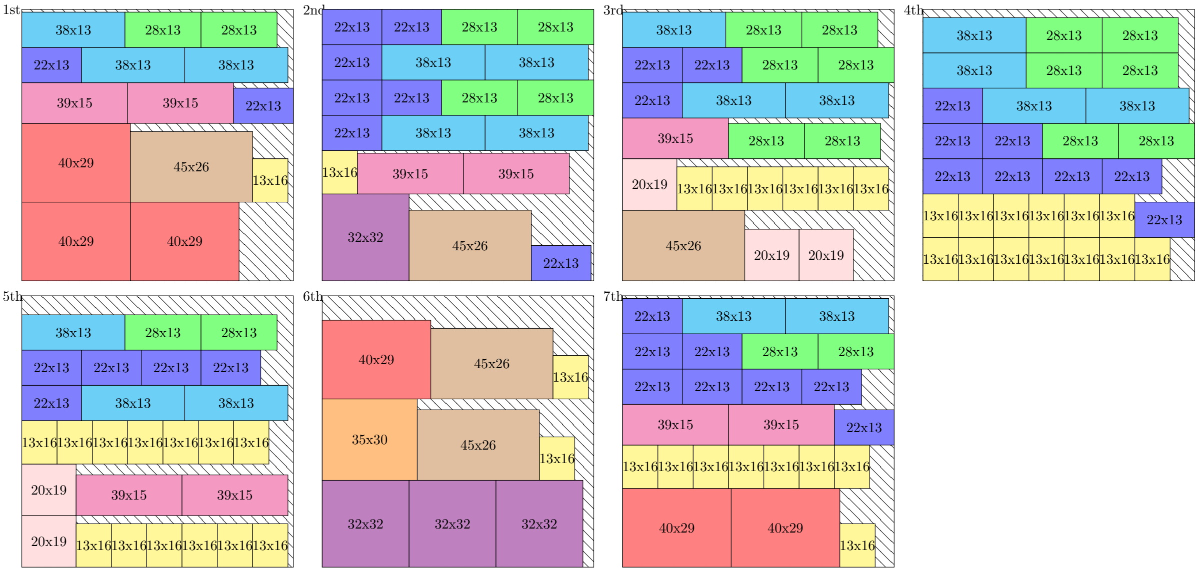
(3) the last row presents averages in columns $\underline{S}$, OFV, LR, gap[%] and time[s], and summations in columns OPT.

Table 6 Number of variables and constraints of the models for the set of instances #B.

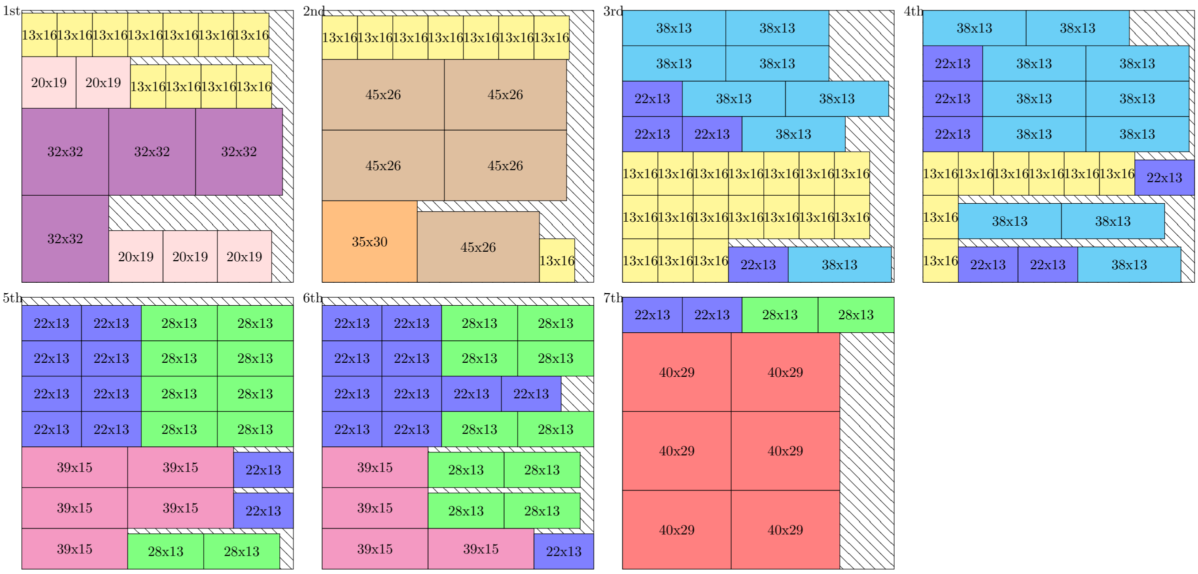
		Model-2stg-IP-HS		Model-2stg-IP		Model-3stg-IP	
		var	cons	var	cons	var	cons
$\overline{M}$=10	$K = 1, 2, 3, 4, 5$	431.36	442.96	2,355.82	826.72	2,646.94	1,423.34
	κ	111.20	23.90	2,036.00	408.00	2,327.30	1,004.80
$\overline{M}$=20	$K = 1, 2, 3, 4, 5$	1,694.24	1,716.64	13,812.94	3,161.04	14,836.36	5,349.66
	κ	429.80	47.90	12,549.10	1,492.90	13,572.70	3,681.70
$\overline{M}$=50	$K = 1, 2, 3, 4, 5$	11,138.68	11,200.02	193,608.62	20,446.46	200,874.30	34,341.76
	κ	2,802.20	126.80	185,275.00	9,376.10	192,539.90	23,270.60

The results in Table 5 show that the average optimality gap of the solver with Model-2stg-IP-HS, Model-2stg-IP, and Model-3stg-IP was 1.01%, 7.60%, and 11.33%, , respectively, with average processing times of 863.55 s, 1,942.20 s and 2,181.58 s, respectively. The average value of the objective function for the Model-2stg-IP-HS, Model-2stg-IP, and Model-3stg-IP was 19.98, 18.39, and 19.26 large objects, respectively. Each entry of the table is an average over ten instances, except those in the last row and columns OPT. The solver was able to find an optimal solution and prove its optimality for 77.22% of the instances (139 out of $180 = 30 \times 6$) instances with Model-2stg-IP-HS, 47.22% (85 instances) with Model-2stg-IP, and 40.00% (72 instances) with Model-3stg-IP. As in the previous section, the proposed models perform better in terms of quality solution and processing time when the number of item types is small ($\overline{M} = 10, 20$) or parameter $K = 1$. Conversely, the results get worse as the patterns become more complex and/or the number of item types $\overline{M}$ increases. The trade-off between the number of required objects and parameter K is similar to that observed in the previous section, where a sharp drop appears when varying $K = 1$ to 2 open stacks. Notice that value of the linear relaxation when $K = 1$ does not correspond to the value of the optimal solution as before in Table 3. In our experiments, we only reached this behavior when $d_i = \lfloor \overline{L}/l_i \rfloor \lfloor \overline{W}/w_i \rfloor$, for $i \in I$. Again, assuming the cutting sequence given by the ordinal set S, for set #B, the average number of simultaneously open stacks was 16.94 for scenarios κ with 92.67 % of the solutions having more than 5 simultaneously open stacks. The average number of item types per object was 2.33 and 4.64 for scenarios $K = 1, 2, 3, 4, 5$ and κ, respectively. In comparison to the results of set #A, these metrics show the relevance of integrating problems, especially when the size of the items relative to the size of the objects decreases.

As the items in set #B are smaller than the items in set #A relative to their objects, the number of variables and constraints of the Model-2stg-IP and Model-3stg-IP almost doubled with instances of set #B. For Model-2stg-IP with $\overline{M} = 50$ and $K = 1, 2, 3, 4, 5$, the number of variables and constraints varied from 98,232.40 and 14,399.27 in Table 4 to 193.608,62 and 20,446.46 in Table 6. This explains the slightly larger average gaps of this section. Note that the number of constraints increases considerably from Model-2stg-IP to Model-3stg-IP. In Fig. 7, we illustrate two solutions for Model-2stg-IP for the instance with $\overline{M} = 10$ item types from set #B. We assumed the scenarios κ and $K = 3$ open stacks. Although both solutions



(a) No sequencing decisions (κ).



(b) $K = 3$.

Figure 7: Illustrations of trade-off solutions for Model-2stg-IP: an instance with $\overline{M} = 10$ of the set #B, assuming no sequencing decisions (κ) and $K = 3$ as the maximum number of simultaneously open stacks
Figure 7 Alt Text: Figure 7a. A solution of non-exact 2-stage patterns with seven objects and nine as the maximum number of open stacks given by scenario κ; Figure 7b. A solution of non-exact 2-stage patterns with seven objects and three as the maximum number of open stacks given by scenario $K = 3$.

require the same number of seven objects to produce the items, the first has the maximum of 9 open stacks at an instant of the cutting sequence, with seven open stacks during the cutting of the first object, and the latter opens just 3 stacks as limited by parameter K. It is interesting to note that in the second solution, all items 32×32 and 20×19 are cut out of the first object, all items 45×26 are cut out of the second object, and all items 40×29 are cut out of the last object.

Our experiments show that satisfactory solutions to the integrated problems can be obtained for a reduced number of open stacks. As a general rule, problem instances of 2D-CSP are harder to solve as the number of item types or their demands increase, and/or the size of the items relative to the size of the objects decreases. As the proposed models are based on the allocation of items to objects, their performance tends to be harmed in terms of quality solution and processing time in these scenarios as well. In the MOSP, the harder problem instances are those whose optimal value K^* of open stacks is considerably less than the number of item types ($K^* \ll \overline{M}$). These scenarios tend to appear in problem instances with sparse matrices like when a few item types are produced from each previously determined pattern. In this sense, the results motivate the development of integrated approaches, since the number of allowed open stacks is considerably less than the number of item types.

6. Conclusions

We addressed three variants of the two-dimensional guillotine cutting stock problem with a limited number of open stacks, as may occur in hardboard and flat glass companies. We referred to these variants as integrated problems (IP), since they integrate the decisions of the cutting stock problem and the minimization of open stacks problems. The three IPs differed with regard to their type of guillotine pattern. The first IP dealt with non-exact 2-stage patterns, the second with exact 2-stage patterns that consider only homogeneous strips, and the last with a specific type of 3-stage patterns. The results of the computational experiments showed that the several equivalent solutions of the cutting problem allowed to obtain satisfactory raw material solutions with a reduced number of open stacks. They pointed out that the integration did not make it easier to solve the models in the context of a branch-and-cut from a general-purpose Integer Linear Programming solver. However, this integration is relevant as approaches that neglect the opening/closing of the stacks can lead to high volumes of work-in-process. Our models are based on the allocation of items to objects and perform best when the number of item types and/or their demands are small.

As future research, one could develop new valid inequalities or elimination of symmetries for the proposed formulations. Another path of research would be the proposition of new formulations for the integrated problems.To solve large problem instances, one could rely on decomposition techniques for integer programming, like Dantzig–Wolfe decomposition, Benders decomposition, and Lagrangean relaxation, or even hybrid methods via matheuristics. Alternative approaches could rely on heuristic and/or meta-heuristic algorithms specifically designed for the problems. Other practical requirements for cutting operations or production scheduling could be considered in the problems, such as other types of patterns or stacks representing customer orders with different item types. We mentioned how to adapt the proposed models to address exact 2-stage patterns. However, these extensions require more research work, as they may not be straightforward for other simpler or more complex patterns, such as p-group, non-staged, non-guillotine, or even for their three-dimensional versions.

Acknowledgements

This work was supported by the São Paulo Research Foundation (FAPESP-Brazil) under Grants [2020/00747-2, 2016/01860-1, 2013/07375-0]; the National Council for Scientific and Technological Development (CNPq-Brazil) under Grant [304144/2019-3]. Research was carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP-Brazil.

Data availability statement

The data that support the findings of this study are available from the corresponding author, [MM], upon reasonable request.

Appendix. Pre-processing operations and lower/upper bounds

As defined before, a problem instance of the 2stg-IP, 2stg-IP-HS, and 3stg-IP is characterized by a tuple $E = (\overline{L}, \overline{W}, \overline{M}, l = (l_1, \ldots, l_{\overline{M}})^\top, w = (w_1, \ldots, w_{\overline{M}})^\top, d = (d_1, \ldots, d_{\overline{M}})^\top, K)$. In what follows, we make use of a reduction method to obtain an equivalent problem instance E' by modifying single input data of problem instance E. First, if $\overline{L} > \overline{L}^*$ with $\overline{L}^* = \max\big\{\sum_{i\in I} l_i a_i : \sum_{i\in I} l_i a_i \leq \overline{L}, a_i \leq d_i, a_i \in \mathbb{Z}_+, i \in I\big\}$, then we decrease the length of the large objects $\overline{L}$ to the value of $\overline{L}^*$. Similarly, if $\overline{W} > \overline{W}^*$ with $\overline{W}^* = \max\big\{\sum_{i\in I} w_i a_i : \sum_{i\in I} w_i a_i \leq \overline{W}, a_i \leq d_i, a_i \in \mathbb{Z}_+, i \in I\big\}$, then we decrease the width of the large objects $\overline{W}$ to the value of $\overline{W}^*$. Solving similar knapsacks that fix at each time n items of type $i \in I$, we can increase the length l_i and width w_i of item types $i \in I$ – see Scheithauer (2018, p. 95) for a detailed description. These two types of pre-processing operations usually provide us an equivalent instance E' with a stronger material bound $\lceil \sum_{i\in I} l_i w_i d_i / (\overline{L}\,\overline{W}) \rceil$. In the 2stg-IP-HS, the knapsacks for increasing the length l_i of item type $i \in I$ can assume $a_k = 0$ for $k \in I \backslash \{i\}$, without loss of optimality, given that this integrated problem deals strictly with homogeneous strips.

As far as the value of lower bound $\underline{S}$ for the 2stg-IP, 2stg-IP-HS, or 3stg-IP is concerned, we solve the following ILP formulation, which is a relaxation for the integrated problems of the cutting decisions. Thus, we define the following variables:

- y^s binary variable which equals 1, if object $s \in S$ is cut, and 0 otherwise;
- x_i^s integer variable that represents the number of items of type $i \in I$ cut out of object $s \in S$.

For this auxiliary problem, we assume $\overline{S} = \sum_{i\in I} d_i$, which is the worst case in which a single item is cut from each object. We denote the value of the (optimal) solution of the following model by LB_1:

$$
\begin{aligned}
\textbf{Min} \quad & \sum_{s \in S} y^s, && \text{(A.1a)} \\
\textbf{s.t.} \quad & (2a) - (2e), && \\
& \sum_{s \in S} x_i^s \geq d_i, & i \in I, \; & \text{(A.1b)} \\
& \sum_{i \in I} (l_i w_i) x_i^s \leq \overline{L}\,\overline{W} y^s, & s \in S, \; & \text{(A.1c)} \\
& y^s \geq y^{j+1}, & s \in S \backslash \{\overline{S}\}, \; & \text{(A.1d)} \\
& x_i^s \leq M^i p_i^s, & s \in S, i \in I, \; & \text{(A.1e)} \\
& y^s \in \{0, 1\}, & s \in S, \; & \text{(A.1f)} \\
& x_i^s \in \mathbb{Z}_+, & s \in S, i \in I. \; & \text{(A.1g)}
\end{aligned}
$$

The objective function (A.1a) minimizes the number of objects required to produce the items. The formulation also relies on the the sequencing problem of constraints (2a) to (2e). Constraints (A.1b) ensure the fulfillment of the demand of the item types. Constraints (A.1c) ensure that the sum of the items' area cut out of object $s \in S$ does not exceed the object's area when $y^s = 1$. Valid inequalities (A.1d) ensure that an object is cut only if its predecessor has also been cut. The linking constraints (A.1e) ensure that, if an item of type $i \in I$ is cut out of object $s \in S$, then its corresponding stack has to be open. Constraints (A.1f) and (A.1f) define the domain of the decision variables. For obtaining the LB_1, we limited the run of the solver to 60 seconds.

We note that the value of LB_1 tends to be a tight lower bound $\underline{S}$ when the value of number K of allowed open stacks is small, such as $K = 1, 2$. For the 2stg-IP and 2stg-IP-HS, we obtain lower bound LB_2 by solving the compact model of Lodi et al. (2004) for the 2D-CSP of 2-stage patterns, considering the complete problem instance. In this case, we limited the run of the solver to 60 seconds. Thus, we define $\underline{S} = \max\{LB_1, LB_2\}$ for the 2stg-IP and 2stg-IP-HS, and $\underline{S} = LB_1$ for the 3stg-IP. Thus, the value of LB_1 provides us a relaxation of the integrated problems for the cutting decisions, and the value of LB_2 a relaxation of the integrated problems concerning the sequencing decisions. For the 2stg-IP-HS, we assume only homogeneous strips when solving the compact model of Lodi et al. (2004) to obtain a tighter lower bound.

We obtain the value of upper bound $\overline{S}$ for the 2stg-IP, 2stg-IP-HS, and 3stg-IP by generating a feasible solution in a constructive framework from model (1) for the 2D-CSP of 2-stage patterns. We first solve this model $\overline{M}$ times, assuming homogeneous patterns only, where each iteration allows the cutting of a single item type $i \in I$. Then, we aggregate these homogeneous cutting patterns contiguously. In other words, we obtain a feasible solution assuming $K = 1$ as the maximum number of simultaneously open stacks. In principle, this strategy generates feasible solutions only for the 2stg-IP and 3stg-IP. However, as mentioned before, we address the 2stg-IP-HS with this model by simply setting variables x_{inj}^s, $s \in S$, $i \in I$, $n \in N_i$, $i < j$, to the value of zero. In the computational experiments of Section 5, in addition to obtaining the value of $\overline{S}$, we provide this generated solution to the solver as an initial solution.

References

Arbib, C., Marinelli, F., & Ventura, P. (2016). One-dimensional cutting stock with a limited number of open stacks: Bounds and solutions from a new integer linear programming model. *International Transactions in Operational Research*, *23*, 47–63. doi:10.1111/itor.12134.

Armbruster, M. (2002). A solution procedure for a pattern sequencing problem as part of a one-dimensional cutting stock problem in the steel industry. *European Journal of Operational Research*, *141*, 328–340. doi:10.1016/S0377-2217(02)00128-5.

Baptiste, P. (2005). Simple MIP formulations to minimize the maximum number of open stacks. In B. M. Smith, & I. P. Gent (Eds.), *Constraint Modelling Challenge 2005*. Edinburgh. URL: https://ipg.host.cs.st-andrews.ac.uk/challenge/.

Beasley, J. E. (1985). Algorithms for Unconstrained Two-Dimensional Guillotine Cutting. *The Journal of the Operational Research Society*, *36*, 297. doi:10.2307/2582416.

Becceneri, J. C., Yanasse, H. H., & Soma, N. Y. (2004). A method for solving the minimization of the maximum number of open stacks problem within a cutting process. *Computers and Operations Research*, *31*, 2315–2332. doi:10.1016/S0305-0548(03)00189-8.

Bennell, J. A., & Oliveira, J. F. (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, *184*, 397–415. URL: https://www.sciencedirect.com/science/article/pii/S0377221706012379. doi:10.1016/j.ejor.2006.11.038.

Carvalho, M. A. M., & Soma, N. Y. (2015). A breadth-first search applied to the minimization of the open stacks. *Journal of the Operational Research Society*, *66*, 936–946. doi:10.1057/jors.2014.60.

Durak, B., & Aksu, D. T. (2017). Dynamic programming and mixed integer programming based algorithms for the online glass cutting problem with defects and production targets. *International Journal of Production Research*, *55*, 7398–7411. doi:10.1080/00207543.2017.1349951.

Dyckhoff, H. (1981). A New Linear Programming Approach to the Cutting Stock Problem. *Operations Research*, *29*, 1092–1104. doi:10.1287/opre.29.6.1092.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, *44*, 145 – 159. URL: http://www.sciencedirect.com/science/article/pii/037722179090350K. doi:http://dx.doi.org/10.1016/0377-2217(90)90350-K. Cutting and Packing.

Dyson, R. G., & Gregory, A. S. (1974). The Cutting Stock Problem in the Flat Glass Industry. *Journal of the Operational Research Society*, *25*, 41–53. doi:10.1057/jors.1974.5.

Frinhani, R. M. D., Carvalho, M. A. M., & Soma, N. Y. (2018). A PageRank-based heuristic for the minimization of open stacks problem. *PLoS ONE*, *13*, 1–24. doi:10.1371/journal.pone.0203076.

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co.

Gilmore, P. C., & Gomory, R. E. (1965). Multistage Cutting Stock Problems of Two and More Dimensions. *Operations Research*, *13*, 94–120. doi:10.1287/opre.13.1.94.

Gonçalves, J. F., Resende, M. G., & Costa, M. D. (2016). A biased random-key genetic algorithm for the minimization of open stacks problem. *International Transactions in Operational Research*, *23*, 25–46. doi:10.1111/itor.12109.

Lemos, F. K., Cherri, A. C., & de Araujo, S. A. (2021). The cutting stock problem with multiple manufacturing modes applied to a construction industry. *International Journal of Production Research*, *59*, 1088–1106. doi:10.1080/00207543.2020.1720923.

Linhares, A. (2004). Minimization of open orders: application of a re-distribution coordination policy. *International Journal of Production Research*, *42*, 1189–1205. doi:10.1080/00207540310001622421.

Linhares, A., & Yanasse, H. H. (2002). Connections between cutting-pattern sequencing, VLSI design, and flexible machines. *Computers and Operations Research*, *29*, 1759–1772. doi:10.1016/S0305-0548(01)00054-5.

Lodi, A., Martello, S., & Monaci, M. (2002). Two-dimensional packing problems: A survey. *European Journal of Operational Research*, *141*, 241–252. doi:10.1016/S0377-2217(02)00123-6.

Lodi, A., Martello, S., & Vigo, D. (2004). Models and Bounds for Two-Dimensional Level Packing Problems. *Journal of Combinatorial Optimization*, *8*, 363–379. doi:10.1023/B:JOCO.0000038915.62826.79.

Lodi, A., & Monaci, M. (2003). Integer linear programming models for 2-staged two-dimensional Knapsack problems. *Mathematical Programming*, *94*, 257–278. URL: http://link.springer.com/10.1007/s10107-002-0319-9. doi:10.1007/s10107-002-0319-9.

Lopes, I. C., & De Carvalho, J. M. (2015). Graph properties of minimization of open stacks problems and a new integer programming model. *Pesquisa Operacional*, *35*, 213–250. doi:10.1590/0101-7438.2015.035.02.0213.

Lucero, S., Marenco, J., & Martínez, F. (2015). An integer programming approach for the 2-schemes strip cutting problem with a sequencing constraint. *Optimization and Engineering*, *16*, 605–632. doi:10.1007/s11081-014-9264-8.

Macedo, R., Alves, C., & Valério de Carvalho, J. M. (2010). Arc-flow model for the two-dimensional guillotine cutting stock problem. *Computers and Operations Research*, *37*, 991–1001. URL: http://dx.doi.org/10.1016/j.cor.2009.08.005. doi:10.1016/j.cor.2009.08.005.

Madsen, O. B. G. (1979). Glass cutting in a small firm. *Mathematical Programming*, *17*, 85–90. doi:10.1007/BF01588227.

Madsen, O. B. G. (1988). An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry. *Journal of the Operational Research Society*, *39*, 249–256. doi:10.1057/jors.1988.42.

Martin, M., Hokama, P. H., Morabito, R., & Munari, P. (2020). The constrained two-dimensional guillotine cutting problem with defects: an ilp formulation, a benders decomposition and a cp-based algorithm. *International Journal of Production Research*, *58*, 2712–2729. doi:10.1080/00207543.2019.1630773.

Martin, M., Morabito, R., & Munari, P. (2021a). Two-stage and one-group two-dimensional guillotine cutting problems with defects: a cp-based algorithm and ilp formulations. *International Journal of Production Research*, *0*, 1–20. doi:10.1080/00207543.2021.1876270.

Martin, M., Yanasse, H. H., & Pinto, M. J. (2021b). Mathematical models for the minimization of open stacks problem. *International Transactions in Operational Research*, *n/a*. doi:https://doi.org/10.1111/itor.13053.

Matsumoto, K., Umetani, S., & Nagamochi, H. (2011). On the one-dimensional stock cutting problem in the paper tube industry. *Journal of Scheduling*, *14*, 281–290. doi:10.1007/s10951-010-0164-2.

Morabito, R., & Arenales, M. (2000). Optimizing the cutting of stock plates in a furniture company. *International Journal of Production Research*, *38*, 2725–2742. doi:10.1080/002075400411457.

Morabito, R., & Belluzzo, L. (2007). Optimising the cutting of wood fibre plates in the hardboard industry. *European Journal of Operational Research*, *183*, 1405–1420. doi:10.1016/j.ejor.2005.11.066.

Parreño, F., & Alvarez-Valdes, R. (2021). Mathematical models for a cutting problem in the glass manufacturing industry. *Omega (United Kingdom)*, *103*. doi:10.1016/j.omega.2021.102432.

Pileggi, G., Morabito, R., & Arenales, M. N. (2005). Abordagens para otimização integrada dos problemas de geração e seqüenciamento de padrões de corte: caso unidimensional. *Pesquisa Operacional*, *25*, 417–447. doi:10.1590/s0101-74382005000300007.

Pileggi, G., Morabito, R., & Arenales, M. N. (2007). Heurísticas para os problemas de geração e sequenciamento de padrões de corte bidimensionais. *Pesquisa Operacional*, *27*, 549–568. doi:10.1590/s0101-74382007000300008.

Puchinger, J., & Raidl, G. R. (2007). Models and algorithms for three-stage two-dimensional bin packing. *European Journal of Operational Research*, *183*, 1304–1327. doi:10.1016/j.ejor.2005.11.064.

Rinaldi, F., & Franz, A. (2007). A two-dimensional strip cutting problem with sequencing constraint. *European Journal of Operational Research*, *183*, 1371–1384. doi:10.1016/j.ejor.2005.12.050.

Scheithauer, G. (2018). *Introduction to cutting and packing optimization: problems, modeling approaches, solution methods*. International Series in Operations Research and Management Science. Springer. doi:10.1007/978-3-319-64143-0.

Silva, E., Alvelos, F., & Valério de Carvalho, J. M. (2010). An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, *205*, 699–708. URL: `http://dx.doi.org/10.1016/j.ejor.2010.01.039`. doi:`10.1016/j.ejor.2010.01.039`.

Smith, B. M., & Gent, I. P. (2005). Constraint modelling challenge 2005. In *The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 05, Edinburgh*.

Valério de Carvalho, J. M. (1999). Exact solution of cutting stock problems using column generation and branch-and-bound. *Annals of Operations Research*, *86*, 629–659. doi:`10.1023/a:1018952112615`.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, *183*, 1109–1130. doi:`10.1016/j.ejor.2005.12.047`.

Yanasse, H. H. (1997a). A transformation for solving a pattern sequencing problem in the wood cut industry. *Pesquisa Operacional*, *17*, 57–70.

Yanasse, H. H. (1997b). On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, *100*, 454–463. doi:`10.1016/S0377-2217(97)84107-0`.

Yanasse, H. H., & Lamosa, M. J. P. (2007). An integrated cutting stock and sequencing problem. *European Journal of Operational Research*, *183*, 1353–1370. doi:`10.1016/j.ejor.2005.09.054`.

Yanasse, H. H., & Morabito, R. (2006). Linear models for 1-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, *44*, 3471–3491. doi:`10.1080/00207540500478603`.

Yanasse, H. H., & Morabito, R. (2008). A note on linear models for two-group and three-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, *46*, 6189–6206. doi:`10.1080/00207540601011543`.

Yuen, B. J. (1991). Heuristics for sequencing cutting patterns. *European Journal of Operational Research*, *55*, 183–190. doi:`10.1016/0377-2217(91)90222-H`.