Two-stage and one-group two-dimensional guillotine cutting problems with defects: a CP-based algorithm and ILP formulations

Mateus Martin[a], Reinaldo Morabito[a,*], Pedro Munari[a]

[a] *Department of Production Engineering, Federal University of São Carlos*

Abstract

We address two variants of the two-dimensional guillotine cutting problem that appear in different manufacturing settings that cut defective objects. Real-world applications include the production of flat glass in the glass industry and the cutting of wooden boards with knotholes in the furniture industry. These variants assume that there are several defects in the object, but the items cut should be defective-free; the cutting pattern is limited to two guillotine stages; and the maximum number of copies per item type in the pattern can be limited. The first variant deals with exact 2-stage patterns, while the second with exact 1-group patterns. To effectively solve these problems, we propose a Constraint Programming (CP) based algorithm as well as different Integer Linear Programming (ILP) formulations. The first presented formulations are extensions of the modeling approach of Martin et al. (2020a) for the case with defects, while the others are novel and more elaborate formulations based on the relative position of the items. We evaluate these three approaches with computational experiments using a set of benchmark instances from the literature. The results show that the approaches find optimal and near-optimal solutions in short processing times for several types of problem instances.

Keywords: Cutting and packing, Guillotine cuts, Defects, 2-stage patterns, 1-group patterns, Constraint Programming, Integer Linear Programming.

1. Introduction

Cutting problems are a class of optimization problems in which a set of items should be cut from a set of objects seeking to minimize or maximize an objective function, e.g., the number of objects used or the sum of the values of the items cut. The arrangement of the items inside the objects (cutting patterns) must prohibit overlapping of items and should generally consider additional geometric restrictions due to the specific characteristics of the cutting machines. These problems arise in different manufacturing settings, such as cutting paper in the paper industry (Matsumoto et al., 2011), flat glass in the glass industry (Durak & Aksu, 2017), and wood in the furniture industry (Morabito & Arenales, 2000; Alem & Morabito, 2012), among many others. In some cases, these cutting problems are integrated with other optimization problems, such as

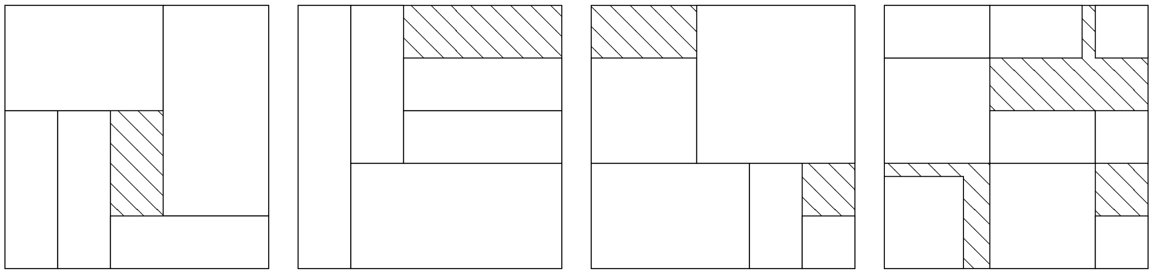
[*] Corresponding author. Department of Production Engineering, Federal University of São Carlos, Via Washington Luiz km. 235, 13565-905, São Carlos-SP, Brazil. Phone/fax: 55-16-33519516/33518240.

Email addresses: `mateus.pmartin@gmail.com` (Mateus Martin), `morabito@ufscar.br` (Reinaldo Morabito), `munari@dep.ufscar.br` (Pedro Munari)

Preprint submitted to Journal

scheduling problems (Hendry et al., 1996), lot-sizing problems (Silva et al., 2014a; Melega et al., 2018) and vehicle routing problems (Iori et al., 2007; Hokama et al., 2016; Alonso et al., 2017).

Two-dimensional cutting problems consider only the length and width for characterizing the shape of the objects and items. The arrangement of the items inside the object (two-dimensional cutting pattern) can be obtained by non-guillotine or guillotine cuts according to the cutting equipment or device. The non-guillotine patterns are usually cut by water jet or laser, while the guillotine patterns are usually cut by cutting saws with edge-to-edge cuts. Hence, when the items are orthogonally cut from an object, a guillotine cut on a rectangle must produce two smaller rectangles, whereas this restriction is not imposed for non-guillotine cuts. Figs. 1a and 1b illustrate non-guillotine and guillotine patterns, respectively, where the items are the blank rectangles and the hatched areas are waste.



(a) non-guillotine. (b) guillotine. (c) 2-stage. (d) 1-group.

Figure 1: Examples of two-dimensional cutting patterns.

Regarding guillotine patterns, there are several manufacturing settings that limit the number of guillotine stages, i.e., the number of 90º rotations of the cutting saw to obtain a higher productivity rate using of shorter cycle times. For instance, in the 2-stage cutting patterns, the horizontal (resp. vertical) cuts of the first stage produce strips, and then the items are obtained from these strips with the vertical (resp. horizontal) cuts of the second stage. Another example is the 1-group cutting patterns, a special case of 2-stage patterns, in which the strips produced by the cuts of the first stage can be stacked and cut together by the cuts of the second stage. In general, 1-group patterns improve the productivity of the cutting machine (e.g., square meters cut per hour) at the expense of increasing the trim loss with respect to general 2-stage patterns (Morabito & Arenales, 2000). Figs. 1c and 1d illustrate, respectively, a 2-stage pattern with a first-stage horizontal cut producing two first-stage horizontal strips and a 1-group pattern with three horizontal and three vertical strips. The solution approaches for the 2-stage cutting problems are often based on mathematical formulations (Lodi et al., 2002; Macedo et al., 2010; Silva et al., 2010; Furini & Malaguti, 2013; Martin et al., 2020a) and column generation algorithms (Gilmore & Gomory, 1965; Belov & Scheithauer, 2006). The 1-group cutting problems are mainly addressed by mathematical formulations (Morabito & Arenales, 2000; Yanasse & Morabito, 2006; Martin et al., 2020a) and enumeration algorithms (Yanasse & Katsurayama, 2005; Yanasse & Morabito, 2008).

All the works mentioned in the previous paragraph considered that the items could be cut from any region of the object (homogeneous object). However, there are manufacturing settings that deal with heterogeneous objects, i.e., the items cannot be obtained from some regions of the objects due to defects. This situation commonly appears when producing customized furniture from wood with knotholes, flat glass ribbon with imperfections due to sulfur, granite and marble with cracks on the stones, to cite a few. There are few approaches in the literature that directly tackle two-dimensional cutting problems with defects. One can cite the approaches based on Dynamic Programming techniques by Hahn (1968) and Scheithauer & Terno (1988) for 3-stage patterns, Carnieri et al. (1993) for objects with a single defect, and Afsharian et al. (2014) for non-stage patterns (i.e., patterns without limitation in the number of guillotine stages). In general, non-stage patterns seek for the highest material utilization rates in detriment of the productivity of the cutting machines. Vianna & Arenales (2006) proposed an implicit enumeration algorithm in a solution space represented by an and/or graph. Recently, Martin et al. (2020b) developed Benders decomposition and Constraint Programming (CP) based algorithms for non-stage patterns with an upper limit on the number of copies per item type (constrained case). The other mentioned works addressed the unconstrained case (i.e., when there is no such upper limit). Please refer to Durak & Aksu (2017) for a discussion on one-dimensional cutting problems with defects and to Gonçalves & Wäscher (2020) for approaches addressing non-guillotine two-dimensional cutting problems with defects. We also note the problem generator of Neidlein et al. (2008) for two-dimensional cutting problems with defects.

In this paper, we address the 2-stage and 1-group two-dimensional guillotine cutting problems with defects. Despite the potential applicability in several industrial environments, to the best of our knowledge, there is no approach in the literature for these two variants. We consider both the unconstrained and constrained cases of these problems and that there may have several overlapping and non-overlapping defects in the object. We focus on exact 2-stage and exact 1-group patterns seeking to improve the productivity of the cutting machine. We also discuss how to adapt the proposed approaches in an approximate manner to deal with non-exact 2-stage and non-exact 1-group patterns. As discussed next in Section 2, exact patterns do not allow an additional guillotine stage to split an item and waste (trimming operation), while non-exact patterns do.

The main contributions of this paper are threefold: (*i*) the development of a CP-based algorithm capable of dealing with 2-stage and 1-group patterns with defects; (*ii*) the proposition of Integer Linear Programming (ILP) formulations based on a discrete approach for these two variants; and (*iii*) the proposition of ILP formulations based on a continuous approach also for the two variants. The CP-based algorithm is a two-phase approach based on the algorithms of Martin et al. (2020b) and Do Nascimento et al. (2019). It first selects a subset of the items to be placed in the object using an ILP model of a one-dimensional cutting problem. Then, in a second phase, it analyzes the non-overlapping of items/defects in a CP model the constraints to generate 2-stage/1-group patterns. In the discrete approach, we extend the models proposed in Martin et al. (2020a) for 2-stage and 1-group patterns to the case with object defects in a straightforward manner. These models are based on allocating items to points of a discretized object taking into account its defects. In the continuous approach, we propose novel models for 2-stage and 1-group patterns with object defects, which are based on the relative position of the items inside the object.

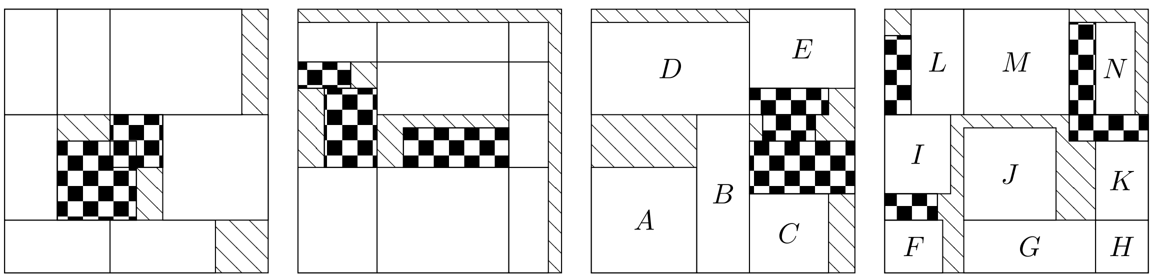
The remainder of this paper is organized as follows. In Section 2, we describe the two addressed variants and present an illustrative example. In Section 3, we develop the CP-based algorithm for both the 2-stage and 1-group cutting problems with defects. In Section 4, we propose the two modeling approaches, namely the discrete and continuous ILP formulations, also for both variants. The computational experiments performed to evaluate the proposed approaches are reported in Section 5. The final remarks and opportunities for future research are presented in Section 6.

2. Problem description

According to the typology of Cutting & Packing problems (Wäscher et al., 2007), we address two cutting problems that can be classified as the Two-dimensional Rectangular Single Large Object Placement Problem (2D_R_SLOPP). The 2D_R_SLOPP consists of selecting and cutting the most valuable subset of items from the object, as an output maximization problem. The two variants we address in this paper consider the following constraints:

- *Geometric constraint.* The cut items must be placed inside the object with their edges parallel to the object's edges, they must not overlap each other, and they must have fixed orientation;
- *Technological constraint.* All cuts must be of guillotine-type, i.e., any cut on a rectangle produces two smaller rectangles, limited to two guillotine stages of exact patterns (trimming operation is not allowed);
- *Production constraint.* There may be an upper limit on the number of copies per item type in the cutting pattern, hence we can address both the unconstrained and constrained cases;
- *Quality constraint.* There are defects in the object. These defects are represented by a set of overlapping rectangles with their positions fixed. The items must be cut from defective-free areas of the object.

The addressed cutting problems differ regarding the type of guillotine cuts. The variant that deals with exact 2-stage patterns is called hereafter 2stg-CPD, while the variant of exact 1-group patterns is called 1grp-CPD. Figs. 2a and 2b illustrate, respectively, cutting patterns of the 2stg-CPD (with three first-stage horizontal strips) and 1grp-CPD (with four horizontal and three vertical strips), where the checkerboard rectangles are the defects. Note that, in order to represent more general cases in practice, the object defects can overlap each other, as depicted in Fig. 2a. Note also that as 2stg-CPD and 1grp-CPD address exact patterns, all the items inside a horizontal strip of Figs. 2a and 2b have the same lower and upper coordinates in the y-axis of a Cartesian plane, while all the items inside a vertical strip of Fig. 2b have the same left and right coordinates in the x-axis of a Cartesian plane. We discuss the non-exact case of Figs. 2c and 2d, i.e., when the trimming operation is allowed, soon after discussing the exact case.



(a) 2stg-CPD. (b) 1grp-CPD. (c) non-exact 2-stage. (d) non-exact 1-group.

Figure 2: Example of 2-stage and 1-group cutting patterns with defects in the exact and the non-exact cases.

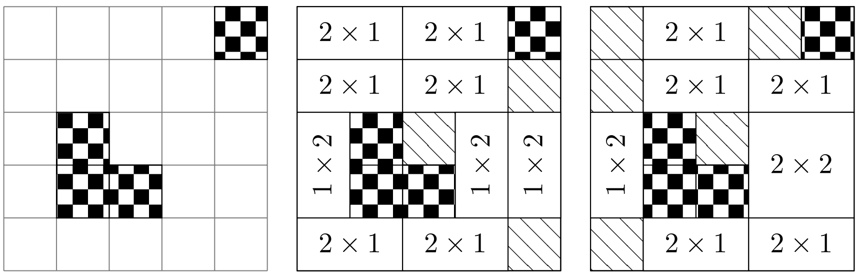
A problem instance of the 2D_R_SLOPP is characterized by the size $L \times W$ of the object and the set $K = \{1, \ldots, m\}$ of item types. Each item type $k \in K$ has size $l_k \times w_k$, value p_k and a maximum number u_k of copies to be produced (Silva et al., 2014b) – we do not consider a minimum number of copies to be produced. Additionally, a problem instance of 2stg-CPD and 1grp-CPD states the set $D = \{1, \ldots, \delta\}$ of defects in the object, where each defect $d \in D$ has size $l_d \times w_d$ and is positioned at (ox_d, oy_d) with respect to the left-lower position of the object (such as the origin of a Cartesian plan).

Table 1 shows an illustrative problem instance of the 2stg-CPD and 1grp-CPD for an object of size 5×5, 3 item types and 3 defects. Regarding this problem instance, Fig. 3 shows the object with the three defects and optimal solutions of the 2stg-CPD (with four first-stage horizontal strips) and the 1grp-CPD (with four horizontal and three vertical strips), respectively. Note that, different from the 2-stage pattern (Fig. 3b), all items of the 1-group pattern (Fig. 3c) are produced using only edge-to-edge cuts on the object. The value of the optimal solution of 2stg-CPD is 51, while this value is 45 for 1grp-CPD. However, the 1grp-CPD pattern is easier to cut in some cutting machines as its horizontal and vertical cuts can be made on the object using the saws without moving each horizontal strip, thus increasing the productivity of the cutting machines (Morabito & Arenales, 2000).

Table 1: A problem instance for an object of size 5×5.

	Item type		
Parameter	1	2	3
l_k	2	1	2
w_k	2	2	1
p_k	10	5	6
u_k	3	4	6

	Defect		
Parameter	1	2	3
ox_d	1	1	4
oy_d	1	1	4
l_d	2	1	1
w_d	1	2	1



(a) object with defects. (b) 2stg-CPD. (c) 1grp-CPD.

Figure 3: The object of the problem instance of Table 1, and optimal solutions for the 2stg-CPD and 1grp-CPD.

Although non-exact 2-stage and non-exact 1-group patterns may yield less waste material if compared to their exact counterparts, the trimming operation required by them may considerably reduce the productivity rate of the cutting process, because it requires an additional cutting step for splitting the items and waste (see Figs. 1c, 1d, 2c and 2d). In the absence of defects, one generally considers, without loss of optimality, that the items are juxtaposed to the left-lower corner, so that all the items inside a horizontal (resp. vertical) strip have the same lower (resp. left) coordinate in y-axis (resp. x-axis) and their upper (resp. right) coordinates may differ for such non-exact patterns, as depicted in Figs. 1c and 1d. If we juxtapose the items to the left-lower corner in the presence of defects, this idea can be kept by considering that the defects are above or to the right of the items in the horizontal and vertical strips, respectively, as depicted by horizontal strip A-B-C (i.e., strip formed by items A, B and C) in Fig. 2c and vertical strips G-J-M and H-K-N in Fig. 2d. However, there is another case in which the defects are below or to the left of the items in the horizontal and vertical strips, respectively, as depicted by item E in the horizontal strip D-E of Fig. 2c and item L in the vertical strip F-I-L of Fig. 2d. Notice that item E is positioned above a defect in the horizontal strip D-E, and both lower and upper coordinates of items D and E in y-axis are not the same. A similar observation applies to item L in the vertical strip F-I-L.

In what follows, we propose approaches for the 2stg-CPD and 1grp-CPD based on exact patterns focusing on the productivity of the cutting machine. As mentioned before, these approaches for exact patterns assume, without loss of generality, that all the items inside a horizontal strip have the same lower and upper coordinates in the y-axis, while all the items inside a vertical strip have the same left and right coordinates in the x-axis. Additionally, we discuss how to adapt these approaches for non-exact patterns in an approximate manner, given that for this case we consider, with loss of generality, that all items inside a horizontal strip have the same lower coordinate in the y-axis and therefore the defects may be only above the items. Moreover, all items inside a vertical strip have the same left coordinate in the x-axis, and therefore the defects may be only to the right of the items.

3. A CP-based algorithm for the 2stg-CPD and 1grp-CPD

In this section, we develop a branch-and-cut algorithm based on CP to solve the 2stg-CPD and the 1grp-CPD. The proposed algorithm decomposes the decisions of these problems into two phases, and it alternates between them until reaching a stopping criterion. The first phase consists of the selection decision of the 2D_R_SLOPP, and the second phase deals with the cutting decision of the 2D_R_SLOPP. The algorithm first uses an ILP model of a one-dimensional cutting problem to find a promising subset of items. Then, in a second phase, it applies a CP model to verify if there is a feasible placement for the subset of items found in the first phase regarding the geometric, technological, production and quality restrictions discussed in Section 2. If the CP model is infeasible (i.e., a feasible placement is not possible for these items), then a combinatorial cut is added to the ILP model of the first phase to eliminate a family of subsets that does not lead to a feasible solution. This type of decomposition algorithm is also known as Logic-based Benders Decomposition (Rahmaniani et al., 2017; Delorme et al., 2017).

The proposed algorithm considers decision variables y_k^n in the ILP model of the first phase, as defined in Equation (1). The formulation is given by Model (2). We note that this model was considered by Martin et al. (2020b) in their CP-based algorithm for non-stage patterns.

$$y_k^n = \begin{cases} 1, & \text{if } n \text{ copies of item type } k \text{ are selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad k \in K, n \in \{0, \dots, u_k\}. \quad (1)$$

$$\textbf{Max} \sum_{k \in K} \sum_{n=0}^{u_k} n p_k y_k^n, \qquad (2a)$$

s.t.

$$\sum_{k \in K} \sum_{n=0}^{u_k} n\, y_k^n\, l_k\, w_k \leq L\,W - \sum_{d \in D} l_d\, w_d, \tag{2b}$$

$$\sum_{n=0}^{u_k} y_k^n = 1, \qquad k \in K, \tag{2c}$$

$$\sum_{k \in K} \sum_{n=\hat{I}(k)}^{u_k} y_k^n \leq m - 1, \qquad \hat{I} \in \mathcal{I}, \tag{2d}$$

$$y_k^n \in \{0, 1\}, \qquad k \in K,\, n \in \{0, \ldots, u_k\}. \tag{2e}$$

Objective function (2a) consists of maximizing the total sum of the values of the selected items. Constraint (2b) ensures that the area of the selected items does not exceed the usable area of the object, given by the area of the object minus the total area of the defects. Constraints (2c) guarantee the production restriction by imposing that only one $n \in \{0, \ldots, u_k\}$ is selected for each item type $k \in K$. Constraints (2d) eliminate all possible families of subsets $\hat{I}$ of items, represented by $\mathcal{I}$, in which there is no feasible placement for all items in $\hat{I}$ into the object. For each $\hat{I} \in \mathcal{I}$, $\hat{I}(k)$ denotes the number of copies of item type $k \in K$ in subset $\hat{I}$. As the number of constraints (2d) is exponential (in the number of items), they are relaxed and removed from the model at first, and then gradually inserted as lazy constraints.

In the first phase of the proposed algorithm, Model (2) is solved by a general-purpose ILP solver with a subset of constraints (2d) composed by those identified in the previous iterations. The optimal solution of this model results in a new subset of items $\hat{I}$. We apply a CP model to verify whether or not there is a feasible placement for this subset $\hat{I}$ into the object. Note that the CP model either finds this feasible placement or proves that it does not exist. In the first case, the algorithm terminates with an optimal solution. In the other case, we create a combinatorial cut of type (2d) using subset $\hat{I}$, which is inserted into Model (2) as a lazy constraint, and a new iteration of the algorithm starts.

The CP model works as follows. Given a subset of items $\hat{I}$ found in the first phase of the algorithm, we define the variables A_i and B_i for each item $i \in \hat{I}$ to indicate the coordinate (A_i, B_i) in which the left-lower point of item i will be allocated. Note that we consider the copies of an item type as different items, i.e., if n copies of item type $k \in K$ are selected, then $\hat{I}$ contains n copies of length l_k and width w_k. By abuse of notation, we use l_i and w_i as the length and width of item $i \in \hat{I}$, respectively. The domain of variables A_i and B_i are defined by sets $\bar{X}$ and $\bar{Y}$. The CP model is given by Model (3).

$$[A_i + l_i \leq A_j] \text{ or } [A_j + l_j \leq A_i] \text{ or } [B_i + w_i \leq B_j] \text{ or } [B_j + w_j \leq B_i], \qquad i \in \hat{I}, j \in \hat{I}, i \neq j, \tag{3a}$$

$$[A_i + l_i \leq ox_d] \text{ or } [ox_d + l_d \leq A_i] \text{ or } [B_i + w_i \leq oy_d] \text{ or } [oy_d + w_d \leq B_i], \qquad i \in \hat{I}, d \in D, \tag{3b}$$

$$([B_i = B_j] \text{ and } [B_i + w_i = B_j + w_j]) \text{ or } [B_i + w_i \leq B_j] \text{ or } [B_j + w_j \leq B_i], \quad i \in \hat{I}, j \in \hat{I}, i \neq j, \tag{3c}$$

$$([A_i = A_j] \text{ and } [A_i + l_i = A_j + l_j]) \text{ or } [A_i + l_i \leq A_j] \text{ or } [A_j + l_j \leq A_i], \qquad i \in \hat{I}, j \in \hat{I}, i \neq j. \tag{3d}$$

Constraints (3a) ensure the non-overlap between two different items $i \in \hat{I}$ and $j \in \hat{I}$, while constraints (3b) ensure the non-overlap between an item $i \in \hat{I}$ and a defect $d \in D$. Constraints (3c) guarantee that two different items $i \in \hat{I}$ and $j \in \hat{I}$ are in the same horizontal strip (i.e., when $B_i = B_j$ and $B_i + w_i = B_j + w_j$) or are above/below each other. These constraints are motivated by those proposed by Do Nascimento et al. (2019) in the context of a CP-based algorithm for non-exact 2-stage patterns of a homogeneous object. For the 1grp-CPD, we also need to consider constraints (3d), which guarantee that two different items $i \in \hat{I}$ and $j \in \hat{I}$ are in the same vertical strip (i.e., when $A_i = A_j$ and $A_i + l_i = A_j + l_j$) or are left/right to each other. Thus, constraints (3c) limit the solution space to exact 2-stage patterns with first-stage horizontal cuts, while (3c)-(3d) limit to exact 1-group patterns. If constraints (3c) are relaxed in Model (3), we have an exact 2-stage pattern with first-stage vertical cuts.

The proposed CP-based algorithm for the 2stg-CPD and 1grp-CPD is summarized in Algorithm 1. As discussed at the end of Section 2, we can adapt, with loss of generality, the CP-approaches for addressing the non-exact case in an approximate manner. This can be done by simply removing $[B_i + w_i = B_j + w_j]$ of constraints (3c) and $[A_i + l_i = A_j + l_j]$ of constraints (3d). As a result, all the items inside a horizontal strip will share the same lower coordinate in y-axis (the upper coordinates will be free), and all the items inside a vertical strip will share the same left coordinate in x-axis (the right coordinates will be free).

Algorithm 1: A CP-based algorithm for the 2stg-CPD and 1grp-CPD.

Input: Instance $(L \times W)$, (l_k, w_k, v_k, u_k) with $k \in K$, and (ox_d, oy_d, l_d, w_d) with $d \in D$.
Output: An optimal solution.

Mount ILP-Model (2) with constraints (2d) relaxed;
repeat
  Find a new subset of items $\hat{I}$ by solving ILP-Model (2);
  Solve CP-Model (3) to find a feasible placement for $\hat{I}$;
  `// Use constraints (3c) for 2stg-CPD or constraints (3c)-(3d) for 1grp-CPD.`
  if *no feasible placement is found* **then**
    Add a combinatorial cut into ILP-Model (2) from subset $\hat{I}$ according to constraint (2d);
until *Optimality is proven;*

The convergence of the algorithm tends to accelerate with improved LP-bounds in the context of an ILP general-purpose solver. Therefore, in a previous version of this study, we also considered relaxations of two-dimensional cutting problems, instead of the one-dimensional cutting problem given by Model (2). In a first attempt, we tried a relaxation of the ILP model proposed in Beasley (1985), and then the contiguous relaxations based model of the Two-dimensional Orthogonal Packing Problem (Scheithauer, 2018). However, the preliminary computational experiments were disappointing in both cases for larger problem instances. As these relaxations are based on the object discretization, they have several symmetrical solutions that had to be eliminated with many combinatorial cuts, which undermine these approaches. The weak LP-bounds of the one-dimensional cutting problem given by Model (2) motivated us to propose the ILP models of the next section, which seem to us more promising alternatives.

4. Models for the 2stg-CPD and 1grp-CPD

In this section, we propose two approaches for modeling the 2stg-CPD and 1grp-CPD. The discrete approach is proposed in Section 4.1, while the continuous approach is proposed in Section 4.2. For both approaches, we start proposing an ILP formulation for the 2stg-CPD, which is then modified for modeling the 1grp-CPD.

4.1. Discrete approach

In what follows, we extend the 2-stage and 1-group models proposed in Martin et al. (2020a), which considered a homogeneous object, in order to consider a heterogeneous object due to defects. These pseudo-polynomial models are based on the discretization of the object as a two-dimensional grid, where the items should be allocated to its points – their number of variables is $O(mLW)$. To consider a defective object, the proposed models prohibit the allocation of items to points that would overlap a defect, similarly to Martin et al. (2020b) in the context of non-stage patterns. We assume the input data as positive integers, without loss of generality. Let set R be the object discretization, as defined in Equation (4). Let set G_k be the points for allocating the left-lower corner of an item type $k \in K$ in which an overlap with any defect is not possible, as defined in Equation (5). Hence, for any point in set G_k, an allocated item type $k \in K$ is completely inside the object, and is positioned to the right, left, above or below in relation to each defect $d \in D$.

$$R = \{(i, j) \in \mathbb{Z}^2 \mid 0 \le i \le L, 0 \le j \le W\}. \tag{4}$$

$$\begin{aligned} G_k = \{&(i, j) \in R \mid i \le L - l_k, j \le W - w_k, \\ &\forall d \in D : (ox_d + l_d \le i) \text{ or } (ox_d \ge i + l_k) \text{ or } (oy_d + w_d \le j) \text{ or } (oy_d \ge j + w_k)\}. \end{aligned} \tag{5}$$

There are two types of decision variables in the formulation for the 2stg-CPD. The first concerns the allocation of an item type k at position (i, j), which is defined in Equation (6). The second is about the horizontal cuts of the first stage, as defined in Equation (7). We discuss later how to adapt the formulation to consider vertical cuts (instead of horizontal cuts) in the first stage.

$$x_{kij} = \begin{cases} 1, & \text{if an item type } k \text{ is allocated at position } (i, j), \\ 0, & \text{otherwise,} \end{cases} \qquad k \in K, (i, j) \in G_k. \tag{6}$$

$$h_j = \begin{cases} 1, & \text{if a horizontal cut is performed from } (0, j) \text{ to } (L, j), \\ 0, & \text{otherwise,} \end{cases} \qquad 0 \le j \le W. \tag{7}$$

As proposed by Beasley (1985) to avoid the overlapping between any pair of allocated items, we consider the binary parameter $f_{kiji'j'}$ that assumes the value of 1 if, and only if, the allocation of item type $k \in K$ at $(i, j) \in G_k$ overlaps point (i', j'), i.e., when $0 \le i \le i' \le i + l_k - 1 < L$ and $0 \le j \le j' \le j + w_k - 1 < W$. An ILP formulation for the 2stg-CPD with first-stage horizontal cuts and based on a discrete approach is given by Model (8).

$$\textbf{Max} \sum_{k \in K} \sum_{(i,j) \in G_k} p_k x_{kij}, \tag{8a}$$

s.t.

$$\sum_{k\in K}\sum_{(i,j)\in G_k} f_{kiji'j'}x_{kij} \leq 1, \qquad (i',j')\in R, i'<L, j'<W, \quad (8b)$$

$$\sum_{(i,j)\in G_k} x_{kij} \leq u_k, \qquad k\in K, \quad (8c)$$

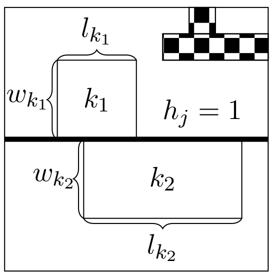
$$\sum_{k\in K}\sum_{(i,j)\in G_k} l_k x_{kij} + \sum_{k\in K}\sum_{\substack{(i,j')\in G_k,\\ j'=j-w_k}} l_k x_{kij'} \leq 2Lh_j, \qquad 0\leq j\leq W, \quad (8d)$$

$$\sum_{k\in K}\sum_{\substack{(i,j')\in G_k,\\ j'<j<j'+w_k}} l_k x_{kij'} \leq L(1-h_j), \qquad 0<j<W, \quad (8e)$$

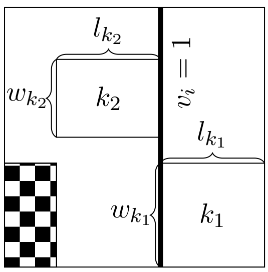
$$x_{kij}\in\{0,1\}, \qquad k\in K, (i,j)\in G_k, \quad (8f)$$

$$h_j\in\{0,1\}, \qquad 0\leq j\leq W. \quad (8g)$$

Objective function (8a) consists of maximizing the total sum of the values of the allocated items. Constraints (8b) ensure the non-overlapping of items by ensuring that at most one allocated item covers each point of the discretization. Constraints (8c) assure the production of at most u_k copies of item type $k \in K$, according to the unconstrained or constrained cases. Constraints (8d) establish that a copy of an item type $k \in K$ can be allocated at $(i,j) \in G_k$ or at $(i, j-w_k) \in G_k$ only if there is a cut in horizontal line j (i.e., when $h_j = 1$), as depicted in Fig. 4a. Constraints (8e) ensure that there is no allocated item that crosses horizontal line j when $h_j = 1$. We highlight that constraints (8d) and (8e) limit the solution space to exact 2-stage patterns of horizontal strips, i.e., with first-stage horizontal cuts. Constraints (8f) and (8g) impose the domain of the variables. Note that constraints (8d) when $h_j = 1$ and constraints (8e) when $h_j = 0$ also work as valid inequalities to ensure that the sum of the allocated items' length in the horizontal line j does not exceed (an order of) the object's length L.



(a) horizontal.



(b) vertical.

Figure 4: Strips for the 2stg-CPD and the 1grp-CPD.

The 1grp-CPD considers patterns formed by horizontal and vertical strips. Therefore, from Model (8), an additional third type of the decision variable is needed related to the vertical cuts of the second stage, as defined in Equation (9). An ILP formulation for the 1grp-CPD based on a discrete approach is given by Model (10).

$$v_i = \begin{cases} 1, & \text{if a vertical cut is performed from } (i,0) \text{ to } (i,W), \\ 0, & \text{otherwise,} \end{cases} \qquad 0 \leq i \leq L. \quad (9)$$

Max (8a),

s.t.

(8b) – (8g),

$$\sum_{k\in K}\sum_{(i,j)\in G_k} w_k x_{kij} + \sum_{k\in K}\sum_{\substack{(i',j)\in G_k,\\ i'=i-l_k}} w_k x_{ki'j} \leq 2Wv_i, \qquad 0 \leq i \leq L, \quad (10a)$$

$$\sum_{k\in K}\sum_{\substack{(i',j)\in G_k,\\ i'<i<i'+l_k}} w_k x_{ki'j} \leq W(1-v_i), \qquad 0 < i < L, \quad (10b)$$

$$v_i \in \{0,1\}, \qquad 0 \leq i \leq L. \quad (10c)$$

Constraints (10a) establish that a copy of an item type $k \in K$ can be allocated at $(i,j) \in G_k$ or at $(i-l_k, j) \in G_k$ only if there is a cut in vertical line i (i.e., when $v_i = 1$), as depicted in Fig. 4b. Constraints (10b) ensure that there is no allocated item that crosses vertical line i when $v_i = 1$. Constraints (10c) state the domain of variables v_i. Similarly to constraints (8d) and (8e), note that constraints (10a) when $v_i = 1$ and constraints (10b) when $v_i = 0$ also work as valid inequalities to ensure that the sum of the allocated items' width in the vertical line i does not exceed (an order of) the object's width W. Note also that when variables h_j and constraints (8d) and (8e) are withdrawn from Model (10), we have an ILP formulation for the 2stg-CPD with first-stage vertical cuts, i.e., with exact 2-stage patterns of vertical strips.

As done in the previous section, we can adapt, with loss of generality, the discrete 2stg-CPD and 1grp-CPD approaches for addressing the non-exact case in an approximate manner. We replace constraints (8d) with constraints (11) in Models (8) and (10), and constraints (10a) with constraints (12) in Model (10). Constraints (11) ensure that all items inside a horizontal strip will share the same lower coordinate in y-axis (the upper coordinates will be free). Constraints (12) ensure that all items inside a vertical strip will share the same left coordinate in x-axis (the right coordinates will be free).

$$\sum_{k\in K}\sum_{(i,j)\in G_k} l_k x_{kij} \leq Lh_j, \qquad 0 \leq j \leq W. \quad (11)$$

$$\sum_{k\in K}\sum_{(i,j)\in G_k} w_k x_{kij} \leq Wv_i, \qquad 0 \leq i \leq L. \quad (12)$$

4.1.1. Discretization of the normal sets

To seek for less variables and constraints in the discrete approach, namely Model (8) for the 2stg-CPD and Model (10) for the 1grp-CPD, we make use of the discretization of the normal sets (Herz, 1972; Christofides & Whitlock, 1977), as discussed in Martin et al. (2020b). In the presence of a defective object, without loss of generality, an allocated item may be to the right or upper side of another allocated item or of a defect. Thus, one could consider only the points that represent positive integer linear combinations of the item types' dimensions and of the defects' positions, instead of a complete discretization of the object. Equations (13) and (14) define these discretization sets for the analyzed problems. The object discretization represented by set R, previously defined in Equation (4), can be redefined using sets $\bar{X}$ and $\bar{Y}$, without loss of generality, as $R = \{(i,j) \in \mathbb{Z}^2 \mid i \in \bar{X}, j \in \bar{Y}\}$. Similarly, set G_k can be redefined according to the new definition of set R.

$$\bar{X} = \left\{ \begin{array}{r} q_x \mid q_x = \sum_{k \in K} n_k l_k + \sum_{d \in D} (ox_d + l_d)\delta_d, \ 0 \leq q_x \leq L, \\ n_k \in \mathbb{N}, n_k \leq u_k, \ \delta_d \in \{0,1\} \end{array} \right\} \cup \{L\}. \tag{13}$$

$$\bar{Y} = \left\{ \begin{array}{r} q_y \mid q_y = \sum_{k \in K} n_k w_k + \sum_{d \in D} (oy_d + w_d)\delta_d, \ 0 \leq q_y \leq W, \\ n_k \in \mathbb{N}, n_k \leq u_k, \ \delta_d \in \{0,1\} \end{array} \right\} \cup \{W\}. \tag{14}$$

The 2stg-CPD and 1grp-CPD consist of exact patterns, and thus the domain of variables h_j (resp. v_i) should contain not only the elements of $\bar{Y}$ (resp. $\bar{X}$), but also $j + w_k$ (resp. $i + l_k$), for any $k \in K$ and $j \in \bar{Y}$ (resp. $i \in \bar{X}$).

4.2. Continuous approach

We present models for 2stg-CPD and 1grp-CPD with polynomial number of binary variables (polynomial in $\overline{n} = \sum_{k \in K} u_k$) in the context of the Padberg-Type model, as called by Scheithauer (2018). In these formulations, we consider the copies of an item type $k \in K$ as different items, i.e., the set $\overline{I} = \{1, \ldots, \overline{n}\}$ contains all the possible items to be produced. To consider a defective object, the proposed models recognize the defects as items allocated at the object with fixed positions and prohibit the overlap of items and defects. Thus, set $I = \{1, \ldots, \overline{n}, \overline{n}+1, \ldots, \overline{n}+\delta\}$ represents all the rectangles, i.e., items and defects.

By abuse of notation, we use l_i and w_i, $i \in I$, as the length and the width of the rectangles. The allocation of the rectangles in the object is expressed by the binary decision variables λ_i, as defined in Equation (15), and the coordinates (α_i, β_i) of the left-lower point of rectangle $i \in I$ are real decision variables, as defined in Equations (16) and (17). To model the non-overlapping of rectangles, we consider the binary variables μ_{ij} and ν_{ij}, as defined in Equations (18) and (19).

$$\lambda_i = \begin{cases} 1, & \text{if rectangle } i \text{ is allocated,} \\ 0, & \text{otherwise,} \end{cases} \qquad i \in I. \tag{15}$$

α_i: position of rectangle i in the x-axis, $\qquad i \in I.$ (16)

β_i: position of rectangle i in the y-axis, $\qquad i \in I.$ (17)

if $\mu_{ij} = 1$, then rect. j is placed right to rect. i (i.e., $\alpha_j \geq \alpha_i + l_i$ holds), $\qquad i \in I, j \in I, i \neq j.$ (18)

if $\nu_{ij} = 1$, then rect. j is placed above rect. i (i.e., $\beta_j \geq \beta_i + w_i$ holds), $\qquad i \in I, j \in I, i \neq j.$ (19)

An ILP formulation for the 2stg-CPD with first-stage horizontal cuts and based on a continuous approach is given by Model (20).

$$\textbf{Max} \sum_{i \in \overline{I}} p_i \lambda_i, \tag{20a}$$

s.t.

$$0 \leq \alpha_i \leq (L - l_i)\lambda_i, \qquad i \in \overline{I}, \tag{20b}$$

$$0 \leq \beta_i \leq (W - w_i)\lambda_i, \qquad i \in \overline{I}, \tag{20c}$$

$$\alpha_i + l_i\lambda_i \leq \alpha_j + L(\lambda_i - \mu_{ij}), \qquad i \in I, j \in I, i \neq j, \tag{20d}$$

$$\beta_i + w_i\lambda_i \leq \beta_j + W(\lambda_i - \nu_{ij}), \qquad i \in I, j \in I, i \neq j, \tag{20e}$$

$$\lambda_i + \lambda_j \leq 1 + \mu_{ij} + \mu_{ji} + \nu_{ij} + \nu_{ji}, \qquad i \in \overline{I}, j \in I, i < j, \tag{20f}$$

$$\mu_{ij} + \mu_{ji} + \nu_{ij} + \nu_{ji} \leq 2\lambda_i, \qquad i \in \overline{I}, j \in I, i < j, \tag{20g}$$

$$\beta_i - \beta_j \leq W(1 - \mu_{ij} - \mu_{ji}) + W(\nu_{ij} + \nu_{ji}), \qquad i \in \overline{I}, j \in \overline{I}, i \neq j, \tag{20h}$$

$$\beta_i + w_i - \beta_j - w_j \leq W(1 - \mu_{ij} - \mu_{ji}) + W(\nu_{ij} + \nu_{ji}), \qquad i \in \overline{I}, j \in \overline{I}, i \neq j, \tag{20i}$$

$$\lambda_i \geq \lambda_{i+1}, \qquad i \in \overline{I}_k \backslash \{\overline{n}\}, i+1 \in \overline{I}_k, \tag{20j}$$

$$\alpha_i, \beta_i \in \mathbb{R}, \qquad i \in I, \tag{20k}$$

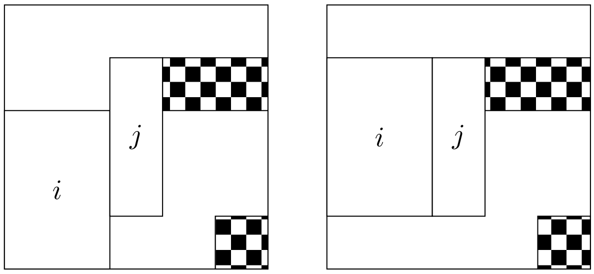
$$\lambda_i \in \{0, 1\}, \qquad i \in I, \tag{20l}$$

$$\mu_{ij}, \nu_{ij} \in \{0, 1\}, \qquad i \in I, j \in I, i \neq j, \tag{20m}$$

$$\lambda_{\overline{n}+d} = 1, \alpha_{\overline{n}+d} = ox_d, \beta_{\overline{n}+d} = oy_d, \qquad d \in D. \tag{20n}$$

Objective function (20a) consists of maximizing the total sum of the values of the allocated items. Constraints (20b) and (20c) ensure that item $i \in \overline{I}$ is completely inside the object when $\lambda_i = 1$, but if $\lambda_i = 0$ then $\alpha_i = \beta_i = 0$ is imposed. Regarding the defects, constraints (20n) work as fixing variables, i.e., they ensure that each defect $d \in D$ is allocated at the object in the corresponding coordinate (ox_d, oy_d). Constraints (20d) and (20e) ensure the definition of variables μ_{ij} and ν_{ij}, respectively. When an item $i \in \overline{I}$ and a rectangle $j \in I$ are allocated (i.e., when $\lambda_i + \lambda_j = 2$), constraints (20f) ensure that they do not overlap by enforcing that at least one of the variables μ_{ij}, μ_{ji}, ν_{ij} and ν_{ji} assume the value 1. Constraints (20g) ensure that all variables μ_{ij} and ν_{ij} related to item i are zero when $\lambda_i = 0$. Constraints (20h) and (20i) limit the solution space to exact 2-stage patterns of horizontal strips. These two blocks of linear constraints can be seen as a linearization of the non-linear constraints (3c). Constraints (20h) ensure that two allocated items are in the same horizontal strip when $\mu_{ij} + \mu_{ji} = 1$ and $\nu_{ij} + \nu_{ji} = 0$, and then $\beta_i = \beta_j$ is enforced. Similarly, constraints (20i) enforce that $\beta_i + w_i = \beta_j + w_j$. Let $\overline{I}_k$ be the set of all items $i \in \overline{I}$ of item type $k \in K$. Constraints (20j) are valid inequalities that limit the allocation of item $i+1$ only when item i is also allocated, if item i and $i+1$ are copies of the same item type. Constraints (20k), (20l) and (20m) impose the domain of the variables. Note that the domain of constraints (20f) allow the overlap of defects over each other, given that $i \in \overline{I}$ (i.e., it varies in the set of items).

As mentioned in the previous section, the 1grp-CPD considers patterns formed by horizontal and vertical strips. Thus, it is also necessary to consider vertical strips to the 1grp-CPD in Model (20). An ILP formulation for the 1grp-CPD based on a continuous approach is given by Model (21).



(a) Without constraints (20h)–(20i).

(b) With constraints (20h)–(20i).

Figure 5: When two allocated items are right/left to and are not below/above each other, constraints (20h)–(20i) ensure that $\beta_i = \beta_j$ and $\beta_i + w_i = \beta_j + w_j$ form a horizontal strip.

Max (20a),

s.t.

(20b) – (20n),

$$\alpha_i - \alpha_j \leq L(1 - \nu_{ij} - \nu_{ji}) + L(\mu_{ij} + \mu_{ji}), \qquad i \in \overline{I}, j \in \overline{I}, i \neq j, \quad (21a)$$

$$\alpha_i + l_i - \alpha_j - l_j \leq L(1 - \nu_{ij} - \nu_{ji}) + L(\mu_{ij} + \mu_{ji}), \qquad i \in \overline{I}, j \in \overline{I}, i \neq j, \quad (21b)$$

Similarly to constraints (20h)–(20i) for items in a horizontal strip, constraints (21a)–(21b) ensure that two allocated items are in the same vertical strip when $\nu_{ij} + \nu_{ji} = 1$ and $\mu_{ij} + \mu_{ji} = 0$, enforcing $\alpha_i = \alpha_j$ and $\alpha_i + l_i = \alpha_j + l_j$. Note that when constraints (20h)–(20i) are withdrawn from Model (21), we have an ILP formulation for the 2stg-CPD with first-stage vertical cuts.

As done before, we can adapt, with loss of generality, the continuous 2stg-CPD and 1grp-CPD approaches for addressing the non-exact case in an approximate manner. It is only necessary to relax constraints (20i) in Models (20) and (21), and relax constraints (21b) in Model (21). Again, as a result, all items inside a horizontal strip will share the same lower coordinate in the y-axis (the upper coordinates will be free), and all items inside a vertical strip will share the same left coordinate in the x-axis (the right coordinates will be free).

To the best of our knowledge, there is no similar continuous modeling approach for 2-stage or 1-group cutting patterns in the literature. Therefore, the present continuous approach is also a novel modeling approach for these cutting patterns in the absence of defects. Therefore, it is only necessary to consider $\delta = 0$, i.e., the set of defects is empty ($D = \varnothing$). Moreover, the extensions of the formulations of Lodi et al. (2002), Silva et al. (2010), Macedo et al. (2010) and Furini & Malaguti (2013) for 2-stage patterns and the formulations of Morabito & Arenales (2000) and Yanasse & Morabito (2006) for 1-group patterns to consider a defective object do not seem to be a straightforward task.

5. Computational experiments

We performed computational experiments to evaluate the quality of solution and processing times of the three proposed approaches for the 2stg-CPD and 1grp-CPD, namely the branch-and-cut algorithm based on CP of Section 3 and the two ILP modeling approaches of Section 4 (discrete and continuous approaches). The CP-based algorithm is referred to as BnCA-2stg and BnCA-1grp, respectively, when addressing the 2stg-CPD and the 1grp-CPD. Models (8) and (10) of the discrete approach are called hereafter Disc-2stg and Disc-1grp, respectively. Models (20) and (21) of the continuous approach are called hereafter Cont-2stg and Cont-1grp, respectively. This section is divided into three parts. In Section 5.1, we describe the sets of instances used in the computational experiments. The results of the proposed approaches for the 2stg-CPD and 1grp-CPD are reported in Section 5.2, while the approximate counterparts of these approaches for the non-exact case are briefly discussed in Section 5.3. All these approaches were coded in C++ using the Concert Studio library on top of IBM CPLEX Optimization Studio 12.8, as an engine for ILP and CP. The experiments were carried out on a PC with Intel Xeon E5-2680v2 (2.8 GHz), limited to 20 threads, 16 GB of RAM, under CentOS Linux 7.2.1511 Operating System. We limited the execution of the solver to 300 seconds which seems to be acceptable in real productive systems. We also used CPLEX to solve ILP-Model (2) with constraints (2d) relaxed, and provided the value of the optimal solution as an upper bound to Disc-2stg, Disc-1grp, Cont-2stg and Cont-1grp.

5.1. Sets of instances

We considered two sets of instances (sets A and B) which are based on the problem instances randomly generated by Afsharian et al. (2014) for the 2D_R_SLOPP with defects for non-stage cutting patterns. These instances are available at `http://www.dep.ufscar.br/docentes/munari/cuttingpacking`. The descriptors and corresponding values used by the authors to generate these instances are as follows:

- size of the object ($L \times W$): (75×75) and (125×50) in category #1, (150×150) and (225×100) in category #2, (300×300) and (450×200) in category #3;
- number of item types (m): 5, 10, 15, 20 or 25;
- size of the item types ($l_k \times w_k$) are taken uniformly from the intervals $[L/\rho, 3L/4]$ and $[W/\rho, 3W/4]$, where ρ is 6, 8 or 10;
- value of the item types is unweighted (i.e., $p_k = l_k w_k$);
- number of defects (δ): 1, 2, 3 or 4;
- size of defects ($l_d \times w_d$) are taken uniformly from the intervals $[L/10, L/6]$ and $[W/10, W/6]$;
- left-lower coordinate of defects (ox_d, oy_d) are taken uniformly from the intervals $[0, L - l_d]$ and $[0, W - w_d]$.

Afsharian et al. (2014) divided these instances into 90 classes ($= 6 \times 5 \times 3$) that consider instances with one to four defects ($= 90 \times 4 = 360$). For each class, they generated 15 instances, but we considered here just the first instance generated for each class in a total of 360 instances. Sets A and B differ regarding the maximum number of copies per item type to be produced (i.e., parameter u_k). Thus, set A considers the unconstrained case with the definition of $u_k = \lfloor L/l_k \rfloor \lfloor W/w_k \rfloor$, for each $k \in K$ (thus u_k is sufficiently large), while set B considers a constrained case with $u_k = \lceil \lfloor L/l_k \rfloor \lfloor W/w_k \rfloor /5 \rceil$, for each $k \in K$.

5.2. *Results for the 2stg-CPD and 1grp-CPD*

We are not aware of any other approach in the literature addressing the 2stg-CPD or 1grp-CPD. Thus, we compare next the proposed approaches with each other in the presence of defects. Tables 2, 3, 4 and 5 report results for the 2stg-CPD and 1grp-CPD, respectively, according to the category of the object's size, number of item types (m), parameter ρ of the item type's size, and number of defects (δ), regarding the problem instances of set A (unconstrained instances) and of set B (constrained instances). Each entry in these tables presents the average relative optimality gap (in percentage) and the average processing time (in seconds) in parenthesis, over all instances in the same category. For each proposed approach and instance, the relative optimality gap is calculated as $(UB - OFV)/(UB + 10^{-10}) * 100$, where UB is the best upper bound obtained by the solver among the three proposed approaches for this instance, and OFV is the value of the best solution found by the solver with the corresponding approach. In these tables, the approach with the best gap among the three proposed approaches is highlighted in bold.

Table 2: Results for the 2stg-CPD and 1grp-CPD according to the category of the object's size (each entry of this table is an average over 120 instances).

	Approach	Category of the object's size		
		#1	#2	#3
Set A	BnCA-2stg	5.56 (184.66)	8.57 (180.25)	10.30 (192.64)
	Disc-2stg	**0.23** (26.93)	**2.38** (73.95)	9.59 (123.18)
	Cont-2stg	3.12 (186.71)	5.77 (196.12)	**6.55** (209.33)
	BnCA-1grp	7.18 (160.56)	12.06 (159.62)	16.74 (177.50)
	Disc-1grp	**0.53** (42.47)	**6.14** (103.69)	18.58 (153.49)
	Cont-1grp	6.08 (176.53)	11.38 (194.56)	**14.86** (207.79)
Set B	BnCA-2stg	1.89 (103.34)	2.71 (96.11)	1.73 (114.08)
	Disc-2stg	**0.60** (60.74)	1.67 (88.14)	7.48 (125.33)
	Cont-2stg	0.69 (34.06)	**0.77** (33.64)	**0.19** (24.34)
	BnCA-1grp	2.12 (80.20)	4.17 (86.37)	2.49 (84.78)
	Disc-1grp	1.73 (119.44)	4.77 (129.47)	11.31 (176.49)
	Cont-1grp	**1.57** (16.74)	**3.16** (47.91)	**1.60** (41.99)

The results presented in Tables 2 and 3 show that, as expected, the greater the category of the object's size or the number of item types, the greater the average gap and the average processing time tend to be. Recall that the number of variables in Disc-2stg and Disc-1grp is $O(mLW)$, while the number of variables in Cont-2stg and Cont-1grp is polynomial in $\overline{n} = \sum_{k \in K} u_k$. Thus, one can see that the average gaps and times of Disc-2stg and Disc-1grp quickly grow as the size $L \times W$ of the object or the number of item types m increases. For instance, in Table 2 for set A the average gap of Disc-2stg is 0.23% (26.93 seconds) in category #1 and 9.59% (123.18 seconds) in category #3, while in Table 3 for set B the average gap of Disc-1grp is 0.00% (8.82 seconds) for $m = 5$ item types and 12.18% (251.55 seconds) for $m = 25$ item types. In Table 2, the results of BnCA-2stg, BnCA-1grp, Cont-2stg and Cont-1grp are less affected by the increase in the object's size than those of Disc-2stg and Disc-1grp. In contrast, for these four approaches in Tables 2 and 3, one can see that their average gaps and times quickly grow as the total number of items $\overline{n}$ or the number of item types m increases. Notice that the total number of items $\overline{n}$ increases as the number of item types m or the maximum number of copies u_k to be produced per item type (e.g., from set B to set A) also increases. For instance, in Table 2 for category #3, the average gap of Cont-1grp is

Table 3: Results for the 2stg-CPD and the 1grp-CPD according to the number of item types (each entry of this table is an average over 72 instances).

	Approach	Number of item types (m)				
		5	10	15	20	25
Set A	BnCA-2stg	**0.00** (7.00)	2.59 (130.54)	9.55 (206.03)	15.06 (290.46)	13.53 (295.23)
	Disc-2stg	**0.00** (4.03)	**0.00** (19.46)	**2.94** (46.12)	12.12 (128.44)	**5.29** (175.38)
	Cont-2stg	**0.00** (48.10)	1.59 (169.15)	4.31 (209.08)	**10.12** (262.10)	9.73 (298.51)
	BnCA-1grp	**0.00** (4.74)	3.31 (87.67)	10.4 (174.36)	20.94 (275.80)	25.31 (286.88)
	Disc-1grp	**0.00** (8.25)	**0.83** (41.02)	**6.09** (67.32)	**18.90** (174.76)	**16.27** (208.07)
	Cont-1grp	**0.00** (39.76)	3.42 (154.16)	9.69 (209.46)	19.19 (262.19)	21.57 (299.21)
Set B	BnCA-2stg	**0.00** (0.27)	**0.00** (9.37)	1.03 (62.93)	3.47 (200.88)	6.05 (249.10)
	Disc-2stg	**0.00** (7.41)	**0.00** (24.71)	1.47 (57.25)	8.99 (154.50)	5.79 (213.14)
	Cont-2stg	**0.00** (1.06)	**0.00** (5.48)	**0.12** (29.71)	**0.20** (37.64)	**2.44** (79.51)
	BnCA-1grp	**0.00** (0.25)	0.94 (20.08)	1.54 (53.66)	3.93 (160.09)	8.22 (184.83)
	Disc-1grp	**0.00** (8.82)	0.74 (79.54)	2.78 (133.18)	13.98 (235.92)	12.18 (251.55)
	Cont-1grp	**0.00** (1.12)	**0.58** (19.75)	**0.38** (26.65)	**2.77** (63.71)	**6.84** (66.49)

1.60% (41.99 seconds) for the constrained instances of set B and 14.86% (207.79 seconds) for the unconstrained instances of set A, while in Table 3 for set A the average gap of BnCA-1grp is 0.00% (4.74 seconds) for $m = 5$ item types and 25.31% (286.88 seconds) for $m = 25$ item types.

Table 4: Results for the 2stg-CPD and the 1grp-CPD according to the item types' size (each entry of this table is an average over 120 instances).

	Approach	Parameter ρ of the item types' size		
		6	8	10
Set A	BnCA-2stg	3.03 (180.01)	9.09 (203.22)	12.32 (174.32)
	Disc-2stg	**0.98** (50.92)	**7.01** (94.26)	**4.22** (78.89)
	Cont-2stg	1.69 (201.79)	7.31 (211.21)	6.44 (179.17)
	BnCA-1grp	4.74 (156.66)	13.77 (176.21)	17.47 (164.80)
	Disc-1grp	**3.15** (72.36)	**13.67** (122.76)	**8.44** (104.53)
	Cont-1grp	4.56 (197.6)	14.02 (201.06)	13.75 (180.21)
Set B	BnCA-2stg	0.33 (81.00)	3.05 (114.24)	2.95 (118.29)
	Disc-2stg	0.46 (61.26)	5.42 (112.74)	3.87 (100.20)
	Cont-2stg	**0.00** (6.31)	**1.00** (38.77)	**0.66** (46.98)
	BnCA-1grp	0.07 (48.08)	4.71 (94.26)	4.00 (109.00)
	Disc-1grp	0.89 (119.88)	10.78 (158.43)	6.13 (147.10)
	Cont-1grp	**0.00** (7.40)	**3.46** (50.92)	**2.88** (48.31)

The results presented in Table 4 show that the average gaps and times increase when parameter ρ varies from 6 to 8 (i.e., the sizes of the item types are smaller compared to the size of the object), as expected. However, this behavior is not observed when parameter ρ varies from 8 to 10 (except in two cases), possibly because the solver with the models obtained better gaps and in less time for $\rho = 10$ than $\rho = 8$, which points out that the solver was able to find better solutions and/or better upper bounds for these instances. Notice that with smaller items, the number of elements in sets $\bar{X}$ and $\bar{Y}$ tends to increase, i.e., the discretization of the object used in Disc-2stg and Disc-1grp

Table 5: Results for the 2stg-CPD and the 1grp-CPD according to the number of defects (each entry of this table is an average over 90 instances).

	Approach	Number of defects (δ)			
		1	2	3	4
Set A	BnCA-2stg	9.42 (196.11)	9.03 (191.64)	6.88 (182.65)	7.25 (173.00)
	Disc-2stg	6.70 (99.15)	4.87 (83.82)	**2.57** (62.34)	**2.14** (53.44)
	Cont-2stg	**4.55** (206.06)	**5.30** (201.93)	5.41 (192.11)	5.35 (189.46)
	BnCA-1grp	13.71 (178.77)	11.29 (170.64)	11.89 (162.51)	11.08 (151.65)
	Disc-1grp	10.97 (120.31)	**9.68** (111.24)	**7.18** (87.97)	**5.85** (80.01)
	Cont-1grp	**10.03** (200.44)	11.33 (194.12)	10.62 (191.58)	11.12 (185.69)
Set B	BnCA-2stg	2.83 (120.15)	1.41 (106.41)	2.20 (95.05)	1.99 (96.42)
	Disc-2stg	7.05 (123.62)	3.16 (105.71)	1.79 (78.12)	1.00 (58.17)
	Cont-2stg	**0.69** (36.11)	**0.40** (28.69)	**0.66** (31.41)	**0.46** (26.51)
	BnCA-1grp	3.53 (91.69)	2.80 (85.31)	2.97 (81.17)	2.41 (76.96)
	Disc-1grp	10.32 (169.47)	6.57 (148.96)	3.85 (138.23)	3.00 (110.55)
	Cont-1grp	**2.80** (38.39)	**1.97** (37.49)	**1.97** (33.49)	**1.72** (32.81)

will have more points (and thus more variables and constraints).

Regarding the number of defects, the results reported in Table 5 show different behaviors among the proposed approaches. For the discrete approaches Disc-2stg and Disc-1grp, the average gaps and times decrease as the number of defects increases, which is mainly explained by the fact that the domain of variables x_{kij} tends to be reduced with more defects. The other four approaches, namely BnCA-2stg, BnCA-1grp, Cont-2stg and Cont-1grp present a similar behavior for the constrained instances of set B. However, for the unconstrained instances of set A, their average gaps and times do not seem to have a direct relation to the number of defects. Thus, these approaches seem more dependent on the total number of items than the number of defects.

In Table 6 we report additional information regarding the proposed approaches aggregated according to the category of the object's size. For the algorithms (i.e., BnCA-2stg and BnCA-1grp), each entry in this table presents the average number of feasible patterns and the average unfeasible patterns (in angle brackets) found by the solver during the search of the corresponding algorithm, over all instances in the same category. For the models (i.e., Disc-2stg, Disc-1grp, Cont-2stg and Cont-1grp), each entry in this table present the average number of variables and the average number of constraints (in square brackets) of the corresponding model, over all instances in the same category. The number of feasible patterns found by the solver with BnCA-2stg and BnCA-1grp is around 5, and more than 2,000 patterns were rejected, which consumes approximately 95% of the processing time of the approaches. Regarding the discrete approaches Disc-2stg and Disc-1grp, the number of variables and constraints are more affected by the category of the object's size than by the total number of items $\overline{n}$ (i.e., from set B to set A). Regarding the continuous approaches Cont-2stg and Cont-1grp, the number of variables and constraints are remarkably affected by the total number of items (i.e., from set B to set A) as they are polynomial in $\overline{n}$. Moreover, the difference in the number of variables of Disc-2stg and Disc-1grp is due to the additional variables v_i used by Disc-1grp for generating the vertical strips of the 1grp-CPD, while the difference in the number of constraints of Cont-2stg and Cont-1grp is due to the additional constraints (21a) and (21b) of Cont-1grp for generating the vertical strips of the 1grp-CPD.

Fig. 6 shows the performance profiles (Dolan & Moré, 2002) based on gaps and times for

Table 6: Average number of variables and constraints for the models, and average number of feasible and unfeasible patterns found by the algorithms (each entry of this table is an average over 120 instances).

		Category of the object's size		
		#1	#2	#3
BnCA-2stg*	Set A	5.03 ⟨2265.93⟩	4.78 ⟨2235.57⟩	4.68 ⟨2121.83⟩
	Set B	5.86 ⟨3379.08⟩	5.29 ⟨3446.17⟩	5.60 ⟨4022.28⟩
BnCA-1grp*	Set A	4.28 ⟨2371.71⟩	3.93 ⟨2214.39⟩	3.95 ⟨2159.37⟩
	Set B	5.13 ⟨3278.58⟩	4.99 ⟨2993.32⟩	5.20 ⟨3125.03⟩
Disc-2stg**	Set A	2166.23 [1163.18]	4231.81 [2426.48]	7808.16 [6365.36]
	Set B	2111.19 [1113.54]	4091.88 [2331.73]	7472.38 [6001.61]
Disc-1grp**	Set A	2217.90 [1265.53]	4316.45 [2594.76]	7942.18 [6632.41]
	Set B	2162.87 [1215.89]	4176.53 [2500.01]	7606.40 [6268.66]
Cont-2stg**	Set A	11926.73 [28838.67]	13538.57 [32830.07]	12998.37 [31453.47]
	Set B	1279.03 [2856.93]	1461.53 [3294.33]	1394.63 [3128.53]
Cont-1grp**	Set A	11926.73 [39913.73]	13538.57 [45482.73]	12998.37 [43542.53]
	Set B	1279.03 [3855.87]	1461.53 [4458.87]	1394.63 [4227.47]

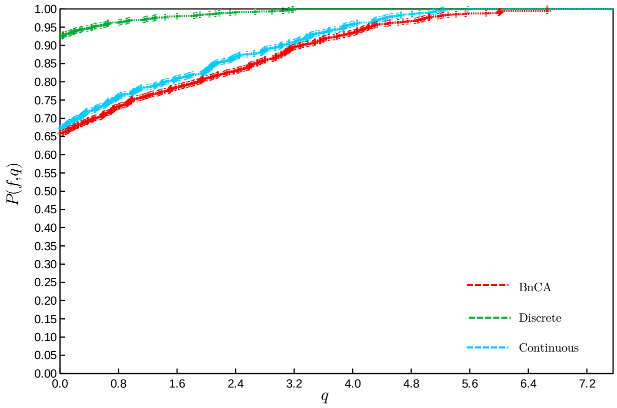
*: each entry shows the average number of "feasible ⟨unfeasible⟩" patterns found by the solver with the corresponding algorithm.

**: each entry shows the average number of "variables [constraints]" of the corresponding model.

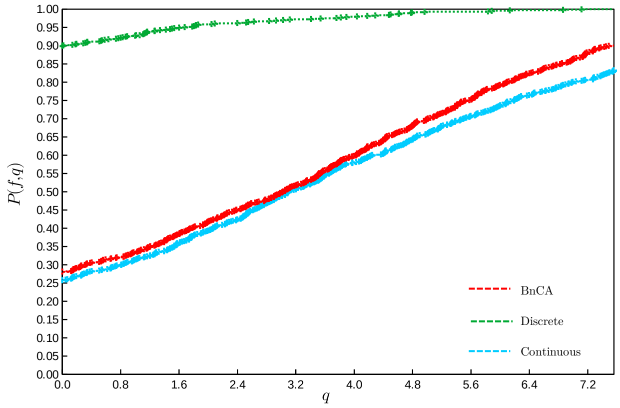
the proposed approaches with the two variants aggregated, i.e., BnCA-2stg and BnCA-1grp are aggregated in BnCA, Disc-2stg and Disc-1grp are aggregated in Discrete, and Cont-2stg and Cont-1grp are aggregated in Continuous. When $q > 0$ in x-axis, the value $P(f, q)$ in y-axis points out the fraction of instances for which the approach f provides solutions with a gap (resp. time) within a factor of 2^q of the best obtained gap (resp. time). When $q = 0$, the value of $P(f, q)$ indicates the fraction of instances for which the approach f reached the best gap (resp. time). For a given instance, the best gap (resp. time) is the lowest gap (resp. time) found considering all the approaches. Clearly, the discrete approach outperforms the BnCA and continuous approaches in the unconstrained case (instances of set A), while the continuous approach outperforms the BnCA and discrete approaches in the constrained case (instances of set B).

We also considered the absence of defects in the object, and then we compared the proposed approaches (adapted to the absence of defects) with the models of Lodi & Monaci (2003) for exact 2-stage patterns and of Yanasse & Morabito (2006) for exact 1-group patterns. As expected, the proposed approaches are not competitive with these benchmark models in the absence of defects. Particularly, the average gaps of the CP-based algorithm (BnCA-2stg and BnCA-1grp), the discrete approach (Disc-2stg and Disc-1grp), and the continuous approach (Cont-2stg and Cont-1grp) were 1.67% (148.97 seconds), 28.67% (117.15 seconds) and 1.85% (117.26 seconds), respectively. In contrast, optimality was proven by the solver with the benchmark models for all instances within an average processing time of just a few seconds. However, as mentioned before, the extensions of these benchmark models to consider a defective object do not seem to be a straightforward task. We also note that the discrete approach is highly benefited by the presence of defects in the object, i.e., while the average gap (time) of the solver with the discrete approach was 28.67% (117.15 seconds) in the absence of defects, much better results were obtained in the presence of defects, as reported in Tables 2, 3, 4 and 5.

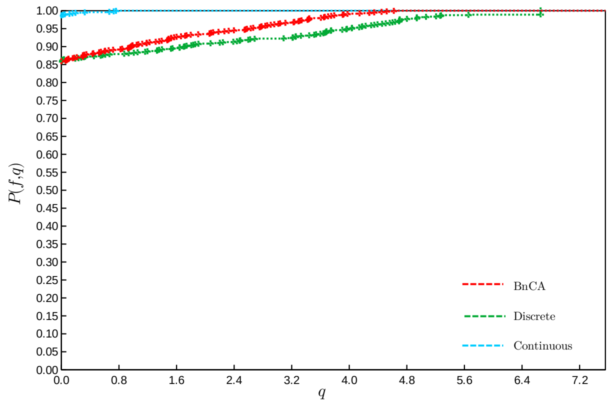
In summary, the main results of this section point out that: (*i*) the discrete approach Disc-2stg and Disc-1grp outperformed the other proposed approaches for the unconstrained instances



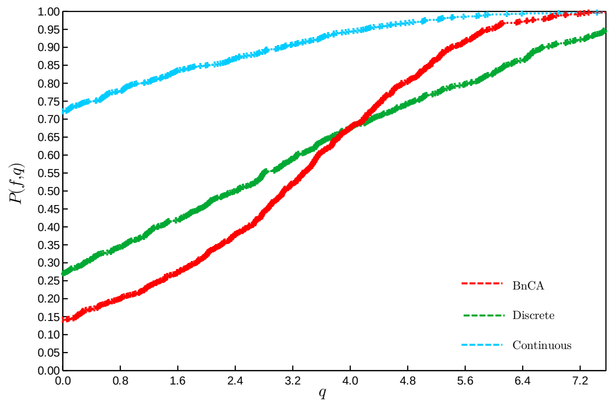
(a) gaps for set A.



(b) times for set A.



(c) gaps for set B.



(d) times for set B.

Figure 6: Performance profiles based on gaps and times for the proposed approaches considering instances of sets A and B. BnCA-2stg and BnCA-1grp are aggregated in BnCA (red), Disc-2stg and Disc-1grp aggregated in Discrete (green), and Cont-2stg and Cont-1grp are aggregated Continuous (blue).

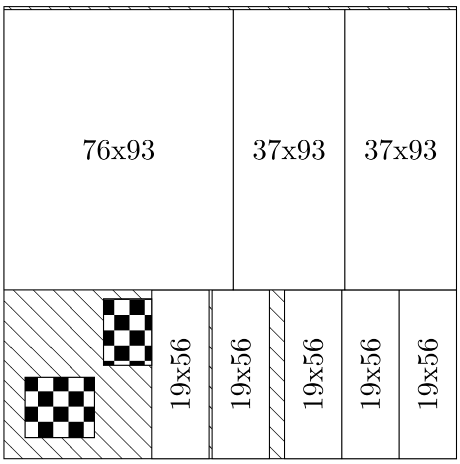
of set A when dealing with small/medium-size objects; (*ii*) the continuous approach Cont-2stg and Cont-1grp outperformed the other proposed approaches for the constrained instances of set B; (*iii*) the quality of the solutions obtained with Disc-2stg and Disc-1grp tends to be dependent on the size of the object and number of item types, while for Cont-2stg and Cont-1grp the quality relates to the total number of items; (*iv*) the proposed algorithms BnCA-2stg and BnCA-1grp are highly dependent on the total number of items, as they outperformed the other approaches only in the instances with $m = 5$ item types (see Table 3) for both sets A and B; and (*v*) the results of the proposed approaches are superior for 2stg-CPD compared to 1grp-CPD, which required more variables/constraints.

5.3. Results for the approximate non-exact versions of the 2stg-CPD and 1grp-CPD

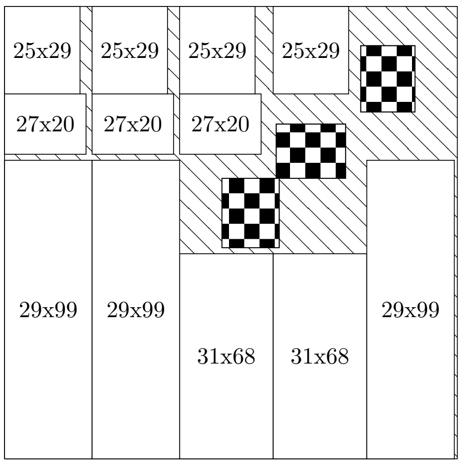
We also performed computational experiments modifying the proposed approaches to cope with the approximate non-exact case, as discussed before. In Table 7, we report results for the 2stg-CPD and 1grp-CPD according to the category of the object's size, regarding the problem instances of set A (unconstrained instances) and of set B (constrained instances), for the approximate non-exact case. These additional results presented similar quality of solution and processing times with respect to the results previously presented in Section 5.2. We note that the proposed approaches led to solutions of better quality more quickly for small instances when adapted to the non-exact case, while they obtained tighter upper bounds for larger problem instances in the exact case. Fig. 7 illustrates solutions of two instances arbitrarily chosen from set A – the exact 2-stage pattern in Fig. 7a was found by the solver with Disc-2stg and is optimal, whereas the non-exact 1-group pattern in Fig. 7b was found by the solver with Disc-1grp modified to deal with the non-exact case (and thus, as discussed before, it is approximate).

Table 7: Results for the approximate non-exact versions of 2stg-CPD and 1grp-CPD according to the category of the object's size (each entry of this table is an average over 120 instances).

	Approach	Category of the object's size		
		#1	#2	#3
Set A	BnCA-2stg	5.38 (196.99)	7.17 (195.80)	8.90 (206.34)
	Disc-2stg	**0.10** (20.22)	**1.42** (64.35)	8.67 (103.83)
	Cont-2stg	2.06 (206.95)	3.76 (217.25)	**5.18** (222.52)
	BnCA-1grp	6.88 (189.98)	8.85 (190.91)	12.50 (203.70)
	Disc-1grp	**0.37** (41.08)	**3.70** (85.64)	11.73 (145.64)
	Cont-1grp	3.56 (199.45)	6.30 (214.10)	**9.63** (218.82)
Set B	BnCA-2stg	1.89 (121.18)	3.80 (130.37)	4.48 (145.94)
	Disc-2stg	**0.24** (25.68)	1.90 (72.58)	8.44 (113.17)
	Cont-2stg	0.47 (76.96)	**1.83** (82.29)	**2.84** (74.95)
	BnCA-1grp	2.42 (119.32)	5.17 (119.46)	5.68 (135.40)
	Disc-1grp	**0.66** (44.33)	5.17 (96.80)	10.87 (158.97)
	Cont-1grp	0.87 (70.10)	**4.02** (81.41)	**4.08** (67.96)



(a) 2stg-CPD.



(b) non-exact version of 1grp-CPD.

Figure 7: Solutions of two instances: (a) (150,150) with $m = 25$ and $\rho = 10$, and (b) (150, 150) with $m = 10$ and $\rho = 10$.

6. Conclusions

We addressed the 2-stage and 1-group Two-dimensional Guillotine Rectangular Single Large Object Placement Problem with Defects. These two variants commonly arise in different manufacturing settings that cut defective objects, for example, in the production of flat glass in the glass industry, in the granite and marble industry with cracks on the stones, in the cutting of wooden boards with knotholes in the furniture industry, among others. In general, the 2-stage patterns yield less waste material but reduce the productivity of the cutting machine (e.g., square meters cut per hour), while the 1-group patterns improve the productivity of the cutting machine with simpler patterns. We focus on exact 2-stage and exact 1-group patterns seeking to improve the productivity of the cutting machine, and we also discuss how to adapt the proposed approaches to non-exact 2-stage and non-exact 1-group patterns in an approximate manner. Despite its potentiality for practical applicability, there are few approaches in the literature that directly tackle cutting problems with defective materials.

We proposed different solution approaches for the two addressed variants. The first is a two-phase algorithm that in the first phase selects a subset of the items to be placed in the object by solving an ILP model of a one-dimensional cutting problem, and then, in the second phase, it analyzes the non-overlapping of items/defects and the constraints to generate 2-stage/1-group patterns by means of a CP model. We also proposed two novel and promising modeling approaches to both variants, one discrete and the other continuous, leading to new ILP models. We are not aware of any other mathematical formulation for these problems in the literature. In the discrete approach, we extended the models proposed in Martin et al. (2020a) for exact 2-stage and exact 1-group patterns to the case with defects in a straightforward manner. These models are based on the allocation of items to points of a discretized object. In the continuous approach, we proposed novel models to 2-stage and 1-group patterns with defects, which are based on the relative position of the items inside the object. Computational experiments performed with benchmark instances of the literature showed that the approaches are able to find optimal and near-optimal solutions in short processing times for several types of problem instances using a general-purpose ILP solver. We highlight the satisfactory results of the discrete models in the unconstrained case, and of the continuous models in the constrained case. The CP-based algorithm seems to be proper for very constrained problem instances and/or with just few items.

For future research, these approaches could be enhanced with the proposition of new valid inequalities to strengthen their linear relaxation, or even to break possible symmetrical solutions. Another possibility is to develop customized solution methods based on decomposition methods for large scale optimization problems from the proposed models, such as Dantzig-Wolfe decomposition, Benders decomposition and Lagrangian relaxation, which could enable the solution of larger problem instances. Regarding the CP-based algorithm, a different decomposition scheme could be sought in order to improve the searches in the master-problem (not only the selection decision) and sub-problem. Regarding the problem themselves, one could analyze the double constrained case (i.e., a lower and a upper limits per item type to be produced), or also adapt the proposed approaches to the version of the problem with multiple defective large objects, for instance, in a column generation framework.

Acknowledgements

The authors would like to thank the National Council for Scientific and Technological Development (CNPq-Brazil) [grant numbers 304601/2017-9, 200745/2018-2] and the São Paulo Research Foundation (FAPESP-Brazil) [grant numbers 2016/08039-1, 2016/01860-1] for the financial support. This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. Research carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP-Brazil [grant number 13/07375-0].

References

Afsharian, M., Niknejad, A., & Wäscher, G. (2014). A heuristic, dynamic programming-based approach for a two-dimensional cutting problem with defects. *OR Spectrum*, *36*, 971–999. doi:10.1007/s00291-014-0363-x.

Alem, D. J., & Morabito, R. (2012). Production planning in furniture settings via robust optimization. *Computers & Operations Research*, *39*, 139 – 150. doi:10.1016/j.cor.2011.02.022.

Alonso, M., Alvarez-Valdes, R., Iori, M., Parreño, F., & Tamarit, J. (2017). Mathematical models for multicontainer loading problems. *Omega*, *66*, 106 – 117. doi:10.1016/j.omega.2016.02.002.

Beasley, J. E. (1985). An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. *Operations Research*, *33*, 49–64. doi:10.1287/opre.33.1.49.

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. doi:10.1016/j.ejor.2004.08.036.

Carnieri, C., Mendoza, G. A., & Luppold, W. G. (1993). Optimal cutting of dimension parts from lumber with a defect. *Forest Produts Journal*, *43*, 66–72.

Christofides, N., & Whitlock, C. (1977). An Algorithm for Two-Dimensional Cutting Problems. *Operations Research*, *25*, 30–44. doi:10.1287/opre.25.1.30.

Delorme, M., Iori, M., & Martello, S. (2017). Logic based benders' decomposition for orthogonal stock cutting problems. *Computers & Operations Research*, *78*, 290 – 298. doi:10.1016/j.cor.2016.09.009.

Do Nascimento, O. X., De Queiroz, T. A., & Junqueira, L. (2019). A MIP-CP based approach for two- and three-dimensional cutting problems with staged guillotine cuts. *Annals of Operations Research*, . doi:10.1007/s10479-019-03466-x.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, *91*, 201–213. doi:10.1007/s101070100263.

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.

Furini, F., & Malaguti, E. (2013). Models for the two-dimensional two-stage cutting stock problem with multiple stock size. *Computers & Operations Research*, *40*, 1953–1962. doi:10.1016/j.cor.2013.02.026.

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., & Wäscher, G. (2020). A MIP model and a biased random-key genetic algorithm based approach for a two-dimensional cutting problem with defects. *European Journal of Operational Research*, . doi:10.1016/j.ejor.2020.04.028.

Hahn, S. G. (1968). On the optimal cutting of defective sheets. *Operations Research*, *16*, 1100–1114. doi:10.1287/opre.16.6.1100.

Hendry, L., Fok, K., & Shek, K. (1996). A cutting stock and scheduling problem in the copper industry. *Journal of the Operational Research Society*, *47*, 38–47. doi:10.1057/jors.1996.4.

Herz, J. C. (1972). Recursive Computational Procedure for Two-dimensional Stock Cutting. *IBM Journal of Research and Development*, *16*, 462–469. doi:10.1147/rd.165.0462.

Hokama, P., Miyazawa, F. K., & Xavier, E. C. (2016). A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications*, *47*, 1 – 13. doi:10.1016/j.eswa.2015.10.013.

Iori, M., Salazar González, J.-J., & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, *41*, 253–264. doi:10.1287/trsc.1060.0165.

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

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

Martin, M., Birgin, E. G., Lobato, R. D., Morabito, R., & Munari, P. (2020a). Models for the two-dimensional rectangular single large placement problem with guillotine cuts and constrained pattern. *International Transactions in Operational Research*, *27*, 767–793. doi:10.1111/itor.12703.

Martin, M., Hokama, P. H., Morabito, R., & Munari, P. (2020b). 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.

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

Melega, G. M., de Araujo, S. A., & Jans, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research*, . doi:10.1016/j.ejor.2018.01.002.

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.

Neidlein, V., Vianna, A. C. G., Arenales, M. N., & Wäscher, G. (2008). *The Two-Dimensional, Rectangular, Guillotineable-Layout Cutting Problem with a Single Defect*. FEMM Working Papers 08035 Otto-von-Guericke University Magdeburg, Faculty of Economics and Management.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*, 801–817. doi:10.1016/j.ejor.2016.12.005.

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.

Scheithauer, G., & Terno, J. (1988). Guillotine cutting of defective boards. *Optimization*, *19*, 111–121. doi:10.1080/02331938808843323.

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. doi:10.1016/j.ejor.2010.01.039.

Silva, E., Alvelos, F., & Valério de Carvalho, J. M. (2014a). Integrating two-dimensional cutting stock and lot-sizing problems. *Journal of the Operational Research Society*, *65*, 108–123. doi:10.1057/jors.2013.25.

Silva, E., Oliveira, J. F., & Wäscher, G. (2014b). 2DCPackGen: A problem generator for two-dimensional rectangular cutting and packing problems. *European Journal of Operational Research*, *237*, 846 – 856. doi:10.1016/j.ejor.2014.02.059.

Vianna, A. C. G., & Arenales, M. N. (2006). O problema de corte de placas defeituosas. *Pesquisa Operacional*, *26*, 185 – 202. doi:10.1590/S0101-74382006000200001.

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., & Katsurayama, D. M. (2005). Checkerboard pattern: proposals for its generation. *International Transactions in Operational Research*, *12*, 21–45. doi:10.1111/j.1475-3995.2005.00488.x.

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.