

An integrated optimization model for the 2-stage two-dimensional cutting stock problem with tardiness penalties *

Kelly Cristina Poldi

Instituto de Matemática, Estatística e Computação Científica (IMECC)
Universidade Estadual de Campinas (UNICAMP)
Rua Sérgio Buarque de Holanda, 651, 13083-859, Campinas-SP, Brasil.
poldi@unicamp.br

Daniel José Schulmeister

Instituto de Matemática, Estatística e Computação Científica (IMECC)
Universidade Estadual de Campinas (UNICAMP)
Rua Sérgio Buarque de Holanda, 651, 13083-859, Campinas-SP, Brasil.
danielschul18@gmail.com

ABSTRACT

This research addresses the Two-Dimensional Stock Cutting and Scheduling Problem (2D-CSSP), which is the combination of the Two-Dimensional Stock Cutting Problem (2D-CSP) and the scheduling problem. This problem, frequently encountered in industrial environments, aims to minimize the sum of terms related to raw material consumption and the maximum and total amount of late periods in fulfilling orders formed by one or more types of items. An integer linear programming model is proposed for the integrated problem, extending a pseudo-polynomial model from the literature developed for the 2D-CSP. An enumeration procedure is presented to obtain 2-stage non-exact guillotine cutting patterns. Computational tests were performed to validate the model using a commercial solver on benchmark instances adapted for the problem.

KEYWORDS. Cutting stock problem. Scheduling problem. Due date. Mathematical formulation. Integer linear programming.

POI – OR in Industry, OD – Discrete Optimization.

*Financial supported by FAPESP 2022/05803-3 and 2024/17600-5.

1. Introduction

The Two-dimensional Cutting Stock and Scheduling Problem (2D-CSSP) corresponds to the integration of the Two-dimensional Cutting Stock Problem (2D-CSP) and scheduling, classified as NP-Hard [Garey and Johnson, 1979], studied extensively in the last decades, and present in various industrial segments. The 2D-CSP deals with the efficient production of rectangular items by cutting larger rectangular objects, so that the a priori known demand for each item type is satisfied, while minimizing an objective function associated with the use of raw material, such as the quantity of objects cut or the total area of material discarded. This mathematical challenge is often addressed in the furniture, glass, paper, and metallurgical industries. In general, the items produced are associated with one or more orders, and each order has a due date. Thus, items from different orders can be geometrically arranged and cut from the same object, which is called a cutting pattern [Gilmore and Gomory, 1965]. The inclusion of due dates in a production plan influences the choice of which cutting patterns should be used and also the order in which they are executed. An order is finalized as soon as the last item that makes it up is produced, and if this occurs after the previously defined due date, then the order is said to be late. In this way, the 2D-CSSP aims to minimize the sum of the costs of cutting and delays in production, both expressed in a convenient monetary unit.

Production is planned to occur over production periods, which can be interpreted as weeks, days, or work shifts. The due date of each order coincides with the end of some existing production period. To quantify the delay in production, the total tardiness of orders (the number of late periods for each order) and/or the maximum tardiness of orders (the maximum violation in relation to due dates) is generally used. Regarding cutting patterns, the following conditions are considered: (i) there cannot be overlapping of items; (ii) the cuts performed are orthogonal guillotine, that is, they are parallel to the edges of the object, giving rise to two smaller rectangles; (iii) up to two stages of guillotine cuts are performed in the cutting pattern and a third stage is allowed for the removal of trimmings; (iv) the orientation of each item is fixed, that is, rotations are not allowed. Such a cutting pattern is classified as 2-stage non-exact guillotine. Figure 1 shows three types of two-dimensional cutting patterns: non-guillotine (1a), obtained by disregarding condition (ii), 2-stage non-exact guillotine (1b) and 2-stage exact guillotine (1c), which satisfy all the mentioned characteristics.

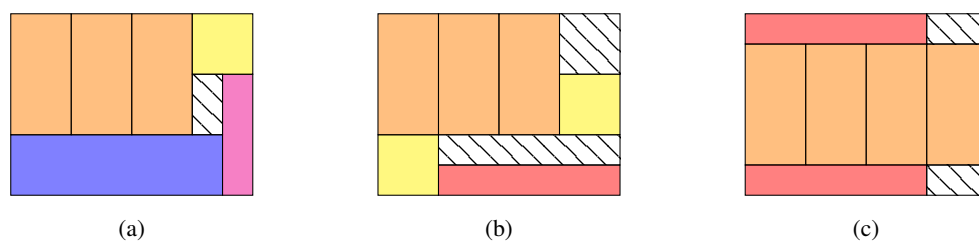


Figure 1: Examples of two-dimensional cutting patterns: (a) non-guillotine, (b) 2-stage non-exact guillotine, and (c) 2-stage exact guillotine. The hatched area represents waste.

In this paper, we propose a new Integer Linear Programming (ILP) model for 2D-CSSP, extending the pseudo-polynomial model proposed for 2D-CSP in [Silva et al., 2010]. We consider, in addition to the quantity of cut objects, the minimization of the weighted sum between the maximum tardiness and the total tardiness of the orders. The developed formulation allows the same item type to be demanded in more than one order throughout the production planning and for orders to be formed by one or more item types, characteristics rarely addressed in the literature.

This paper is organized as follows: Section 2 presents a brief literature review on the problem. Section 3 describes the problem mathematically and proposes an integer linear programming

model. An item-object enumeration procedure for generating 2-stage non-exact guillotine cutting patterns is described in Section 3.1. Section 4 presents computational results obtained with instances from the literature adapted for the problem, and in Section 5, final conclusions are reported.

2. Literature review

In the literature, some papers deal with the integration of the cutting stock problem with scheduling. In [Li, 1996], a model is proposed that associates previously known two-dimensional cutting patterns to periods of arbitrary size, tackled with LP-based and non-LP-based heuristics. The one-dimensional version is addressed in [Reinertsen and Vossen, 2010], with the proposition of an exact model only when the Earliest Due Date (EDD) rule holds. A few years later, [Arbib and Marinelli, 2014] consider the same problem, describes a model for associating cutting patterns to periods, and develops a resolution procedure based on period refinement, column generation, and a partial enumeration scheme.

The classical model of [Kantorovich, 1960] is extended with the notions of sequencing in [Braga et al., 2016] and reinforced with knapsack constraints generated with dual-feasible functions. This formulation is considered in [Pitombeira-Neto and Prata, 2020], covering the case where orders are formed by several types of items. On the other hand, the pseudo-polynomial arc-flow model proposed by [Valério de Carvalho, 1999] is extended in [Braga et al., 2015], considering the minimization of the total tardiness of orders and unit periods. To solve the problem, the aggregation of consecutive periods and a post-processing to minimize the production delay are proposed. In [Oliveira et al., 2025], an ILP model is proposed to minimize the tardiness and earliness of orders and machine setup. A procedure composed of column generation, relax-and-fix heuristic, and a local search is developed as a solution methodology.

The Two-Dimensional Bin Packing Problem (2D-BPP), the version of 2D-CSP with unitary demand items, is also addressed in the literature, integrating it with production sequencing, as explained in [Bennell et al., 2013; Polyakovskiy and M'Hallah, 2017; Arbib et al., 2021], in which the maximum lateness of orders is minimized. In [Martin et al., 2024], the objective is to minimize the total spread and maximum spread of orders, where the spread is given by the number of periods between the start and end of processing of each order.

3. Problem description and proposed model

Consider a production schedule divided into T production periods and denote by $\mathcal{T} = \{1, \dots, T\}$ the set of corresponding periods and $\mathcal{P} = \{1, \dots, P\}$ the set of requested orders. The rectangular objects available in each period have lengths L and widths W . Let $\mathcal{I} = \{1, \dots, I\}$ be the set of item types, of length ℓ_i and width w_i . The demand for item type i in order p is represented by b_{ip} , $i \in \mathcal{I}$, $p \in \mathcal{P}$, and the due date of each order is given by $d_p \in \mathcal{T}$.

Orders composed of more than one item type are said to be heterogeneous, and the corresponding demand vector is given by $\mathbf{b}_p = \{b_{1p}, \dots, b_{Ip}\}$. Orders for only one item type are said to be homogeneous, and the demand vector has only one non-zero component. Denote by $\mathcal{P}^i$ the set of orders $p \in \mathcal{P}$ in which item type $i \in \mathcal{I}$ has non-zero demand.

The ILP model proposed as follows extends the formulation developed for the 2-stage guillotine 2D-CSP in [Silva et al., 2010], considering production periods and orders with due dates. The model described in [Silva et al., 2010], in turn, extends the *one-cut* model, introduced by [Dyckhoff, 1981], which is based on the notion of cutting, so that when an object of length L is cut to obtain an item of length ℓ_i , a residual object of length $L - \ell_i$ is obtained and becomes available for cutting.

In the 2-stage two-dimensional case, considering that the guillotine cuts of the first stage are horizontal, each time an object of dimensions $L \times W$ is cut to obtain an item of dimensions $\ell_i \times w_i$, two residual rectangles are obtained, the first of dimensions $(L - \ell_i) \times w_i$ and the second

of dimensions $L \times (W - w_i)$. In the first rectangle, a horizontal cut (first stage) can be performed again, as long as its dimensions allow some item to be obtained. On the other hand, in the second rectangle, only the second stage (vertical) cuts can be performed. Since no item can be cut from a residual rectangle, it is disregarded. In this way, an enumeration of all possible item-object pairs is previously calculated, according to Algorithm 1 (see Section 3.1).

Figure 2 illustrates the steps of the enumeration algorithm for the case where the original object has dimensions 4×5 , item type 1 (orange color) 3×2 , and item type 2 (yellow color) 1×1 . Figure (2a) shows that the action of cutting one item type 1 from the original object gives rise to two residual objects: R_1 of dimensions 4×3 and R_2 of dimensions 1×2 . Figure (2b) shows that the action of cutting one item type 1 from the residual object R_1 produces the residual objects R_3 and R_4 of dimensions 4×1 and 1×2 , respectively.

In Figure (2c), one item type 2 is cut from the residual object R_2 and it is observed that the obtained residual object R_5 , of dimensions 1×1 , must be discarded, since in the residual object R_2 only second-stage cuts can be performed (in other words, in the residual object R_5 no item can be allocated, otherwise the cutting pattern would not be 2-stage). Note that $R_2 = R_4$, therefore (2c) may happen twice. Figure (2d) presents a final feasible 2-stage cutting pattern composed of item types 1 and 2.

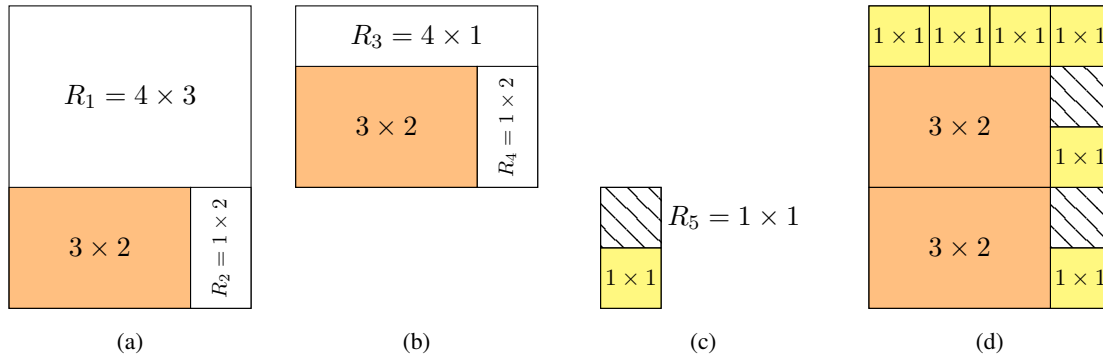


Figure 2: Examples of the process of cutting items from an object. In (a) and (b), item type 1 (orange color) is cut from the original object and from the residual object, respectively. In (c), item type 2 (yellow color) is cut from a residual object. A feasible cutting pattern is presented in (d). The hatched area represents waste.

Let $\mathcal{K} = \{0, 1, \dots, K\}$ be the set of residual object types, where object type 0 is the original object. The model has four decision variables. The non-negative integer variable x_{ik}^t indicates the number of item type $i \in \mathcal{I}$ cut from object type $k \in \mathcal{K}$ in period $t \in \mathcal{T}$. The non-negative integer variable z_{ip}^t , which corresponds to the number of item types $i \in \mathcal{I}$ produced for order $p \in \mathcal{P}$ in period $t \in \mathcal{T}$. The binary variable y_p^t , which is equal to 1 if the order $p \in \mathcal{P}$ was not completed in the period $t = \{d_p, \dots, T\}$, and 0 otherwise. Finally, the non-negative integer variable Y represents the maximum tardiness between orders, that is, the worst violation of due dates.

The parameter a_{ikh} is defined as 1 if, by cutting the item type i in the object type k , we obtain the object type h . Furthermore, for each object type $k \in \mathcal{K}$, the set $\mathcal{Q}^k$ is formed by the item type $i \in \mathcal{I}$ that can be cut in the object type k . Similarly, for each item type $i \in \mathcal{I}$, the set $\mathcal{V}^i$ gathers the object type k in which the item type i can be cut.

The proposed model is defined as follows:

$$\min \omega_0 \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} x_{i0}^t + \omega_1 Y + \omega_2 \sum_{p \in \mathcal{P}} \sum_{t > d_p} y_p^t, \quad (1)$$

$$\text{s.t.: } \sum_{k \in \mathcal{V}^i} x_{ik}^t = \sum_{p \in \mathcal{P}^i} z_{ip}^t, \quad i \in \mathcal{I}, t \in \mathcal{T}, \quad (2)$$

$$\sum_{t \in \mathcal{T}} z_{ip}^t = b_{ip}, \quad i \in \mathcal{I}, p \in \mathcal{P}^i, \quad (3)$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{V}^i} a_{ijk} x_{ij}^t \geq \sum_{i \in \mathcal{Q}^k} x_{ik}^t, \quad k \in \mathcal{K} \setminus \{0\}, t \in \mathcal{T}, \quad (4)$$

$$z_{ip}^t \leq O_{ip} y_p^t, \quad i \in \mathcal{I}, p \in \mathcal{P}^i, t > d_p, \quad (5)$$

$$y_p^{t+1} \leq y_p^t, \quad p \in \mathcal{P}, t > d_p, \quad (6)$$

$$\sum_{t > d_p} y_p^t \leq Y, \quad p \in \mathcal{P}, \quad (7)$$

$$\sum_{i \in \mathcal{Q}^0} x_{i0}^t \leq C_t, \quad t \in \mathcal{T}, \quad (8)$$

$$x_{ik}^t \in \mathbb{Z}_+, \quad i \in \mathcal{I}, k \in \mathcal{V}^i, t \in \mathcal{T}, \quad (9)$$

$$Y \in \mathbb{Z}_+, \quad (10)$$

$$z_{ip}^t \in \mathbb{Z}_+, \quad i \in \mathcal{I}, p \in \mathcal{P}^i, t \in \mathcal{T}, \quad (11)$$

$$y_p^t \in \{0, 1\}, \quad p \in \mathcal{P}, t > d_p. \quad (12)$$

The objective function (1) minimizes the sum of the quantity of (original) objects cut, maximum tardiness between orders, and total tardiness, according to the parameters ω_0, ω_2 , and ω_1 , respectively. Such parameters are responsible for a monetary conversion of each term of the objective function. The constraints (2) and (3) guarantee the production of the items demanded in each order. The first set ensures that the quantity of item type $i \in \mathcal{I}$ produced in each period $t \in \mathcal{T}$ is distributed to one or more orders in which item type i is required. The second set imposes that the demand for each item type in order $p \in \mathcal{P}$ is satisfied by considering all production periods. In (4), it follows that for each period $t \in \mathcal{T}$ the number of cut objects type $k \in \mathcal{K} \setminus \{0\}$ cannot be greater than the number of objects obtained of the same type. The correct counting of periods in which an order $p \in \mathcal{P}$ is late is done by means of the inequalities (5) and (6). In (5), it is ensured that the production of item type $i \in \mathcal{I}$ for order $p \in \mathcal{P}^i$ in period $t \in \mathcal{T}$ occurs only if the order was not completely processed, that is, if $y_p^t = 1$. The parameter O_{ip} is an upper bound for the maximum quantity of item type $i \in \mathcal{I}$ produced in order $p \in \mathcal{P}$. The inequalities (6) establish that if order $p \in \mathcal{P}$ was not completely processed in the period $t > d_p$, then the same is true for the previous periods. The maximum violation of due dates is defined in (7). The quantity of (original) objects cut in each period $t \in \mathcal{T}$ cannot be greater than the production capacity of the period, as pointed out in (8). Finally, in (9) to (12) the domains of the model's decision variables are defined.

3.1. Item-object pair enumeration procedure

As mentioned earlier, it is necessary to perform an enumeration of all feasible item-object pairs previously. Algorithm 1 describes this procedure, based on the algorithms developed in [Silva et al., 2010] (for the 2- and 3-stage cases) and in [Furini et al., 2016] (for the non-staged case).

The algorithm starts by including the triple $(L, W, 1)$ in the set of rectangles R , corresponding to the original object. The first two components of the triple correspond to the dimensions of the object, while the third indicates the stage of the cut that can be made on the object. This rectangle, indicated by the index 0, is marked as unprocessed. The process is finished as soon as all rectangles in R have been processed.

Then, an unprocessed rectangle r in R is selected, and the item type $i \in \mathcal{I}$ that has dimensions such that it can be cut into the rectangle is stored in $\mathcal{Q}^r$ (steps 3-6). For each item type

$i \in \mathcal{Q}^r$, the two residual rectangles resulting from the cut of the item in the rectangle r are defined: R_1 , of dimensions $L_r \times (W_r - w_i)$, and R_2 , of dimensions $(L_r - \ell_i) \times w_i$. The third component of the triples related to these rectangles tells which stage can be executed in each rectangle (steps 9-10). Figure 3 illustrates obtaining R_1 and R_2 .

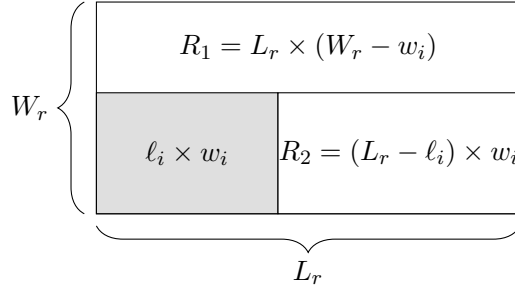


Figure 3: Residual objects R_1 and R_2 generated after cutting one item of dimensions $\ell_i \times w_i$ from the object of dimensions $L_r \times W_r$.

If on rectangle r only second-stage cuts can be performed (i.e., r is a rectangle of type R_2), then when analyzing the residual rectangles generated by cutting some item in r , every rectangle of type R_1 must be discarded. If it were possible to allocate any item in the rectangle of type R_1 , then the resulting cutting pattern would not be 2-stage, but rather 3 or more stages. For this reason, rectangles of type R_1 can only be obtained from an object r on which first-stage cuts can be performed (step 11).

If no item can be cut from R_1 , then the rectangle is discarded. Otherwise, it is checked whether R_1 has already been obtained in previous steps. If not, the triple associated with R_1 is included in R , R_1 is marked as unprocessed and the parameter $a_{i,r,|R|} = 1$ is set. This parameter means that cutting the item type i in the object r results in the residual object with index $|R|$, where $|R|$ is the number of elements in the set R (after including the triple associated with R_1). On the other hand, if the rectangle R_1 has already been included in R , it is sufficient to set the parameter $a_{i,r,\lambda} = 1$, where λ is the index of the rectangle in R equal to R_1 (steps 12-19). The analysis for R_2 is analogous (steps 20-27). Finally, the rectangle r is marked as processed (step 28).

4. Computational results

In this section, computational experiments are reported to evaluate the proposed ILP model. All implementations were coded in Julia 1.9.2/JuMP 1.12.0 with the solver Gurobi 12.0.1, using a computer with an Intel(R) Core(TM) i7-8700 CPU 3.20GHz, with 32Gb of RAM, and Windows 10 operating system. The tolerance established was 10^{-8} and the time limit for each run of the solver was limited to 600 seconds.

The set of instances used is an adaptation of the *gcutd1-12* instances proposed in [Cintra et al., 2008] for 2D-CSP. These, in turn, were obtained by adding a random demand between $[0, 100]$ to the *gcutd1-12* instances proposed in [Beasley, 1985], generated for the Two-dimensional Knapsack Problem (2D-KP). The dimensions $L \times W$ of the objects are 250×250 in *gcutd1-4*, 500×500 in *gcutd5-8* and 1000×1000 in *gcutd9-12*. The number of item types I varies in 10, 20, 30, and 50. The dimensions $\ell_i \times w_i$ of each item type $i \in \mathcal{I}$ were sampled in the intervals $[L/4, 3L/4]$ and $[W/4, 3W/4]$.

In order to obtain instances for the 2D-CSSP, the number of periods (T), production capacity in each period (C_t), number of orders (P), demand for item type i in order p (b_{ip}) and the due date of each order (d_p) must be specified. For each parameter, the following considerations are made.

Algorithm 1: Item-object enumeration for 2-stage non-exact guillotine cutting patterns

Input : Length L and width W of the original object and set of item types $\mathcal{I}$

Output: Set of residual objects R and parameter a_{ijk}

```

1 Initialize the set  $R$  with the triple  $(L_0 = L, W_0 = W, est_0 = 1)$ 
2 Mark rectangle 0 as unprocessed;
3 while  $R$  contains an unprocessed rectangle do
4   Select an unprocessed rectangle  $r \in R$ 
5    $Q^r \leftarrow \emptyset$ 
6   for  $i \in \mathcal{I}$  do
7     if  $w_i \leq W_r$  and  $\ell_i \leq L_r$  then  $Q^r = Q^i \cup \{r\}$ 
8   for  $i \in Q^r$  do
9      $R_1 \leftarrow (L_r, W_r - w_i, 1)$ 
10     $R_2 \leftarrow (L_r - \ell_i, w_i, 2)$ 
11    if  $est_r = 1$  then
12      if some item type  $i \in \mathcal{I}$  can be allocated in  $R_1$  then
13        if  $R_1 \notin R$  then
14           $R \leftarrow R \cup \{R_1\}$ 
15          Mark  $R_1$  as unprocessed
16          Set  $a_{i,r,|R|} \leftarrow 1$ 
17        else
18          Let  $\lambda$  be the index of the rectangle such that  $R_1 = R_\lambda$ 
19          Set  $a_{i,r,\lambda} \leftarrow 1$ 
20    if some item type  $i$  can be allocated in  $R_2$  then
21      if  $R_2 \notin R$  then
22         $R \leftarrow R \cup \{R_2\}$ 
23        Mark  $R_2$  as unprocessed
24        Set  $a_{i,r,|R|} \leftarrow 1$ 
25      else
26        Let  $\lambda$  be the index of the rectangle such that  $R_2 = R_\lambda$ 
27        Set  $a_{i,r,\lambda} \leftarrow 1$ 
28  Mark  $r$  as processed
  
```

- The number of periods are chosen arbitrarily as $T = 5, 10$, and 15 .
- Each period has the same production capacity ($C_1 = \dots = C_T$). To determine it, the number of objects needed to satisfy the total demand for items is divided by the quantity of objects T , and the integer greater than or equal to it is taken. These quantities can be extracted from [Furini and Malaguti, 2013], or obtained by solving the 2D-CSP integrated in the model (1) - (12).
- Homogeneous and heterogeneous orders are handled separately. In the homogeneous case, the total quantity of orders is $P = I \cdot T$, so that there are T orders for each item type $i \in \mathcal{I}$. The demand for item type $i \in \mathcal{I}$ in order $p \in \mathcal{P}^i$ is generated by randomly dividing the demand for item type i in the *gcutd* instance by T . Finally, it is established that the t -th order

for each item type has the same due date, which is randomly generated in the interval $[1, T]$.

- For the heterogeneous case, the total quantity of orders is $P = T$. The demand for item type $i \in \mathcal{I}$ in order $p \in \mathcal{P}$ is equal to the demand for the t -th order for item type i , in the case of homogeneous orders (previous item). Basically, the t -th order for each item type is grouped together in a single order. The same due dates as in the previous case are preserved.

In this way, the same data set can be used for cases of homogeneous and heterogeneous orders, changing only the number of orders for each case. Thus, there are 36 instances (12 for $T = 5, 10, 15$), referred to below as “*gcutds-T-L-I*” and available in (add repository after acceptance).

The problem is solved by considering three different scenarios, each characterized by the presence (or absence) of each of the three terms that form the objective function (1). The first scenario focuses only on minimizing the number of cut objects without taking into account any scheduling constraints, imposing $\omega_0 = 1, \omega_1 = \omega_2 = 0$. In the second scenario, it is also desired to minimize the sum of the tardiness of each request, while maintaining the same number of cut objects as in the first scenario. Thus, the following difference in the order of magnitude between the parameters is established: $\omega_0 = T^2, \omega_1 = 0, \omega_2 = 1$. The last scenario adds the minimization of the maximum tardiness between the orders, so that this has a greater impact on the solution than the sum of the tardiness of the orders. To keep the same amount of cut objects, $\omega_0 = T^3, \omega_1 = T^2, \omega_2 = 1$ is set.

Table 1 shows the computational results for each of the scenarios described above, considering only homogeneous orders. The first column provides information on the name and the averages for each set of 12 instances ($T = 5, 10, 15$). The second column ($\sum x$) reports the number of objects cut in the 2D-CSP solution associated with the problem. The other columns are divided according to the three scenarios. For each group, the maximum tardiness among the orders (Y), the sum of the tardiness of each request ($\sum y$), the execution time in seconds (time[s]), and the gap in percentage (gap[%]) are displayed.

It can be observed that for $T = 5$, optimality was reached for all instances, with average execution times of 0.39s, 20.15s, and 44.72s in the first, second, and third scenarios, respectively. For $T = 10$, all instances were resolved to optimality in the first scenario, 10 instances in the second scenario, and 11 in the third (average gap less than 0.01%), with average execution times of 1.23s, 179.36s, 191.96s. Finally, for $T = 15$, all instances were resolved to optimality in the first scenario, 4 in the second, and 5 in the third (average gap less than 0.01%), with average execution times of 2.66s, 480.52s, and 434.98s. The results indicate that the inclusion of the terms referring to the maximum lateness of orders and the sum of the lateness of orders in the objective function imply a greater computational effort, as expected. Furthermore, the same behavior is observed when increasing the number of production periods.

However, the most interesting result is obtained when analyzing the variation of the maximum lateness and the sum of the lateness of the orders in each scenario. The first scenario indicates that, among the multiple optimal solutions that minimize the number of cut objects, due to the absence of the terms for minimizing maximum lateness and sum of lateness of orders in the objective function, there are some solutions with these metrics reaching relatively large values, when compared to the other scenarios. For $T = 5, 10$, and 15, the maximum lateness reduced, from the second scenario to the third, by 32%, 45%, and 60%, respectively, and the sum of the lateness of the orders increased by 17%, 19%, and 44%, respectively. Thus, it is clear that the inclusion of the minimization of the maximum lateness, in addition to reducing the maximum lateness, implies an increase in the sum of the lateness of the orders.

The columns of Table 2 are organized in the same format as Table 1, and show the compu-

Table 1: Results of the computational tests performed on instances with homogeneous orders.

Instance name	$\sum x$	$\omega_0 = 1, \omega_1 = \omega_2 = 0$				$\omega_0 = T^2, \omega_1 = 0, \omega_2 = 1$				$\omega_0 = T^3, \omega_1 = T^2, \omega_2 = 1$			
		Y	$\sum y$	time[s]	gap[%]	Y	$\sum y$	time[s]	gap[%]	Y	$\sum y$	time[s]	gap[%]
<i>gcutds-5-250-10</i>	295	4	55	0.18	0.00	1	1	0.10	0.00	1	1	0.07	0.00
<i>gcutds-5-250-20</i>	345	4	174	0.15	0.00	4	29	46.57	0.00	3	29	15.14	0.00
<i>gcutds-5-250-30</i>	342	2	95	0.68	0.00	2	4	9.43	0.00	1	5	13.30	0.00
<i>gcutds-5-250-50</i>	845	4	447	0.68	0.00	4	51	87.48	0.00	2	67	434.95	0.00
<i>gcutds-5-500-10</i>	207	4	67	0.08	0.00	1	2	0.17	0.00	1	2	0.22	0.00
<i>gcutds-5-500-20</i>	375	3	82	0.19	0.00	2	3	0.51	0.00	1	4	0.73	0.00
<i>gcutds-5-500-30</i>	600	4	185	0.18	0.00	4	15	1.31	0.00	2	17	5.30	0.00
<i>gcutds-5-500-50</i>	720	3	144	1.12	0.00	1	1	2.86	0.00	1	1	8.73	0.00
<i>gcutds-5-1000-10</i>	135	3	47	0.09	0.00	1	2	0.24	0.00	1	2	1.72	0.00
<i>gcutds-5-1000-20</i>	315	3	95	0.18	0.00	2	4	0.48	0.00	2	4	1.54	0.00
<i>gcutds-5-1000-30</i>	349	4	239	0.45	0.00	4	54	65.95	0.00	3	62	45.66	0.00
<i>gcutds-5-1000-50</i>	675	4	247	0.71	0.00	2	2	26.73	0.00	1	2	9.28	0.00
Average	433.58	3.50	156.42	0.39	0.00	2.33	14.00	20.15	0.00	1.58	16.33	44.72	0.00
<i>gcutds-10-250-10</i>	295	9	260	0.07	0.00	8	28	24.36	0.00	3	37	12.61	0.00
<i>gcutds-10-250-20</i>	345	9	548	0.44	0.00	8	51	tl	<0.01	3	61	429.65	0.00
<i>gcutds-10-250-30</i>	342	9	439	1.36	0.00	1	8	343.67	0.00	1	8	314.81	0.00
<i>gcutds-10-250-50</i>	845	7	667	1.71	0.00	3	14	190.67	0.00	2	16	367.02	0.00
<i>gcutds-10-500-10</i>	207	9	283	0.16	0.00	3	17	7.52	0.00	2	19	3.90	0.00
<i>gcutds-10-500-20</i>	375	8	284	0.38	0.00	2	9	9.31	0.00	1	11	4.32	0.00
<i>gcutds-10-500-30</i>	600	8	446	0.90	0.00	3	8	20.68	0.00	2	8	9.73	0.00
<i>gcutds-10-500-50</i>	720	7	580	5.52	0.00	1	2	328.84	0.00	1	2	505.11	0.00
<i>gcutds-10-1000-10</i>	135	9	204	0.11	0.00	6	19	11.54	0.00	3	23	32.31	0.00
<i>gcutds-10-1000-20</i>	315	9	297	0.61	0.00	3	13	9.20	0.00	2	21	19.62	0.00
<i>gcutds-10-1000-30</i>	349	8	310	1.55	0.00	1	1	6.22	0.00	1	1	4.26	0.00
<i>gcutds-10-1000-50</i>	675	9	716	2.00	0.00	3	15	tl	<0.01	2	14	tl	<0.01
Average	433.58	8.42	419.50	1.23	0.00	3.50	15.42	179.36	<0.01	1.92	18.42	191.96	<0.01
<i>gcutds-15-250-10</i>	295	14	504	0.13	0.00	11	52	tl	<0.01	5	62	114.02	0.00
<i>gcutds-15-250-20</i>	345	14	451	2.32	0.00	1	1	8.21	0.00	1	1	6.34	0.00
<i>gcutds-15-250-30</i>	342	13	926	5.28	0.00	5	21	tl	0.01	2	30	tl	<0.01
<i>gcutds-15-250-50</i>	845	14	1778	3.19	0.00	4	8	596.83	0.00	2	17	tl	<0.01
<i>gcutds-15-500-10</i>	207	14	468	0.51	0.00	9	41	253.32	0.00	4	42	425.22	0.00
<i>gcutds-15-500-20</i>	375	14	760	0.82	0.00	11	63	tl	<0.01	4	77	tl	<0.01
<i>gcutds-15-500-30</i>	600	14	992	1.64	0.00	3	9	106.03	0.00	1	14	284.93	0.00
<i>gcutds-15-500-50</i>	720	13	1570	7.45	0.00	8	31	tl	<0.01	2	32	tl	<0.01
<i>gcutds-15-1000-10</i>	135	13	362	0.68	0.00	6	33	tl	<0.01	3	52	187.30	0.00
<i>gcutds-15-1000-20</i>	315	14	768	1.51	0.00	13	108	tl	<0.01	4	191	tl	<0.01
<i>gcutds-15-1000-30</i>	349	13	995	3.96	0.00	7	35	tl	0.01	2	50	tl	<0.01
<i>gcutds-15-1000-50</i>	675	14	1767	4.36	0.00	14	93	tl	0.01	6	145	tl	0.04
Average	433.58	13.67	945.08	2.66	0.00	7.67	41.25	480.52	<0.01	3.00	59.42	434.98	<0.01

tational results for each scenario, considering only heterogeneous orders. All instances were solved to optimality for $T = 5$, with average execution times of 0.38s, 4.91s, 1.85s, and for $T = 10$, with average execution times of 1.21s, 67.96s, and 29.85s. For $T = 15$, all instances were solved to optimality in the first scenario, 8 in the second (average gap less than 0.01%), and 9 in the third (average gap less than 0.01%), with average execution times of 2.44s, 268.52s, and 230.33s. It is observed that there was less computational effort in relation to execution time to solve instances with heterogeneous orders compared to instances with homogeneous orders.

Regarding the variation of the maximum lateness and the sum of the lateness of the orders, it follows that for $T = 5, 10$ and 15 , the maximum lateness reduced, from the second scenario to the third, by 27%, 46%, and 51%, respectively, and the sum of the lateness of the orders increased by 15%, 10%, and 3%, respectively.

5. Conclusions

The Two-dimensional Cutting Stock and Scheduling Problem was addressed in this work. An integer linear programming model was proposed, extending the model developed for 2D-CSP in [Silva et al., 2010], considering homogeneous and/or heterogeneous orders. The possibility of a type of item being demanded in more than one order was included, together with an objective function that minimizes the sum of three terms: the quantity of objects used, the maximum lateness between orders, and the sum of the lateness of each order.

The computational tests were performed with instances adapted from the literature. The results were analyzed by distinguishing three different scenarios, defined by the inclusion (or not) of each term that composed the objective function. Most instances were solved until optimality, and the others presented a gap lower than 0.01%. The results indicate that it is possible to obtain optimal solutions in relation to the number of objects used in each scenario and, in addition, to find solutions that minimize the sum of the lateness of the orders and the maximum lateness between the orders. The parameters $\omega_0, \omega_1, \omega_2$ can be used to define a priority among the terms to be minimized and also to capture price variations in the value of the raw material and possible fines related to the late delivery of orders in practical situations.

6. Acknowledgements

The authors would like to thank the São Paulo Research Foundation (FAPESP-Brazil) [grant numbers 2022/05803-3 and 2024/17600-5] for the financial support.

References

- Arbib, C. and Marinelli, F. (2014). On cutting stock with due dates. *Omega*, 46:11–20.
- Arbib, C., Marinelli, F., and Pizzuti, A. (2021). Number of bins and maximum lateness minimization in two-dimensional bin packing. *European Journal of Operational Research*, 291(1): 101–113.
- Beasley, J. E. (1985). Algorithms for unconstrained two-dimensional guillotine cutting. *The Journal of the Operational Research Society*, 36(4):297–306.
- Bennell, J. A., Lee, L. S., and Potts, C. N. (2013). A genetic algorithm for two-dimensional bin packing with due dates. *International Journal of Production Economics*, 145:547–560.
- Braga, N., Alves, C., Macedo, R., and Valério de Carvalho, J. M. (2015). A model-based heuristic for the combined cutting stock and scheduling problem. In *Computational Science and Its Applications – ICCSA 2015*, p. 490–505, Cham. Springer International Publishing.

Table 2: Results of the computational tests performed on instances with heterogeneous orders.

Instance name	$\sum x$	$\omega_0 = 1, \omega_1 = \omega_2 = 0$				$\omega_0 = T^2, \omega_1 = 0, \omega_2 = 1$				$\omega_0 = T^3, \omega_1 = T^2, \omega_2 = 1$			
		Y	$\sum y$	time[s]	gap[%]	Y	$\sum y$	time[s]	gap[%]	Y	$\sum y$	time[s]	gap[%]
<i>gcutds-5-250-10</i>	295	4	9	0.04	0.00	1	1	0.07	0.00	1	1	0.09	0.00
<i>gcutds-5-250-20</i>	345	4	15	0.15	0.00	4	6	0.95	0.00	3	8	1.82	0.00
<i>gcutds-5-250-30</i>	342	2	7	0.68	0.00	2	2	2.86	0.00	1	2	2.82	0.00
<i>gcutds-5-250-50</i>	845	4	15	0.68	0.00	4	7	36.48	0.00	2	9	6.25	0.00
<i>gcutds-5-500-10</i>	207	4	11	0.06	0.00	1	1	0.13	0.00	1	1	0.22	0.00
<i>gcutds-5-500-20</i>	375	3	8	0.18	0.00	2	2	0.31	0.00	1	2	0.20	0.00
<i>gcutds-5-500-30</i>	600	4	11	0.21	0.00	3	5	1.07	0.00	2	5	2.71	0.00
<i>gcutds-5-500-50</i>	720	3	6	1.08	0.00	1	1	4.07	0.00	1	1	2.47	0.00
<i>gcutds-5-1000-10</i>	135	3	8	0.10	0.00	1	2	0.29	0.00	1	2	0.15	0.00
<i>gcutds-5-1000-20</i>	315	3	10	0.19	0.00	2	2	0.20	0.00	2	2	0.50	0.00
<i>gcutds-5-1000-30</i>	349	4	16	0.48	0.00	4	8	4.62	0.00	3	10	3.36	0.00
<i>gcutds-5-1000-50</i>	675	4	10	0.69	0.00	1	2	7.91	0.00	1	2	1.56	0.00
Average	433.58	3.50	10.50	0.38	0.00	2.17	3.25	4.91	0.00	1.58	3.75	1.85	0.00
<i>gcutds-10-250-10</i>	295	9	55	0.07	0.00	8	11	3.74	0.00	3	13	1.29	0.00
<i>gcutds-10-250-20</i>	345	9	62	0.46	0.00	5	14	26.44	0.00	3	14	5.85	0.00
<i>gcutds-10-250-30</i>	342	9	39	1.04	0.00	2	5	43.50	0.00	1	5	39.17	0.00
<i>gcutds-10-250-50</i>	845	7	41	1.52	0.00	2	4	68.53	0.00	2	4	68.01	0.00
<i>gcutds-10-500-10</i>	207	9	53	0.15	0.00	4	8	6.92	0.00	2	8	6.59	0.00
<i>gcutds-10-500-20</i>	375	8	43	0.38	0.00	2	3	3.22	0.00	1	3	2.15	0.00
<i>gcutds-10-500-30</i>	600	8	42	0.89	0.00	4	4	6.73	0.00	2	5	7.43	0.00
<i>gcutds-10-500-50</i>	720	7	34	5.65	0.00	2	2	524.02	0.00	1	2	148.35	0.00
<i>gcutds-10-1000-10</i>	135	9	48	0.12	0.00	6	10	9.45	0.00	3	12	7.95	0.00
<i>gcutds-10-1000-20</i>	315	9	41	0.61	0.00	4	4	4.17	0.00	2	5	3.46	0.00
<i>gcutds-10-1000-30</i>	349	8	31	1.57	0.00	1	1	2.59	0.00	1	1	2.95	0.00
<i>gcutds-10-1000-50</i>	675	9	38	2.04	0.00	3	3	116.22	0.00	2	4	65.00	0.00
Average	433.58	8.42	43.92	1.21	0.00	3.58	5.75	67.96	0.00	1.92	6.33	29.85	0.00
<i>gcutds-15-250-10</i>	295	14	126	0.13	0.00	11	25	298.13	0.00	5	29	11.21	0.00
<i>gcutds-15-250-20</i>	345	14	89	1.88	0.00	1	1	3.18	0.00	1	1	1.80	0.00
<i>gcutds-15-250-30</i>	342	13	99	5.30	0.00	4	7	tl	<0.01	2	6	tl	<0.01
<i>gcutds-15-250-50</i>	845	14	97	3.39	0.00	3	6	212.18	0.00	2	6	159.74	0.00
<i>gcutds-15-500-10</i>	207	14	107	0.46	0.00	4	18	31.86	0.00	4	19	tl	<0.01
<i>gcutds-15-500-20</i>	375	14	113	0.56	0.00	5	17	37.19	0.00	4	17	44.98	0.00
<i>gcutds-15-500-30</i>	600	14	97	1.21	0.00	2	6	82.72	0.00	1	6	23.48	0.00
<i>gcutds-15-500-50</i>	720	13	99	7.30	0.00	2	7	tl	<0.01	2	5	tl	<0.01
<i>gcutds-15-1000-10</i>	135	13	101	0.63	0.00	6	10	25.21	0.00	3	12	16.89	0.00
<i>gcutds-15-1000-20</i>	315	14	125	0.99	0.00	11	31	tl	<0.01	4	35	77.47	0.00
<i>gcutds-15-1000-30</i>	349	13	107	3.36	0.00	7	9	130.91	0.00	2	9	147.01	0.00
<i>gcutds-15-1000-50</i>	675	14	110	4.02	0.00	10	23	tl	<0.01	2	19	480.82	0.00
Average	433.58	13.67	105.83	2.44	0.00	5.50	13.33	268.52	0.00	2.67	13.67	230.33	0.00

- Braga, N., Alves, C., Macedo, R., and Valério de Carvalho, J. M. (2016). Exact solution of combined cutting stock and scheduling problems. In *Computational Management Science*, p. 131–139, Cham. Springer International Publishing.
- Cintra, G. F., Miyazawa, F. K., Wakabayashi, Y., and Xavier, E. C. (2008). Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191(1):61–85.
- Dyckhoff, H. (1981). A new linear programming approach to the cutting stock problem. *Operations Research*, 29:1092–1104.
- Furini, F. and Malaguti, E. (2013). Models for the two-dimensional two-stage cutting stock problem with multiple stock size. *Computers & Operations Research*, 40(8):1953–1962.
- Furini, F., Malaguti, E., and Thomopulos, D. (2016). Modeling two-dimensional guillotine cutting problems via integer programming. *INFORMS Journal on Computing*, 28:736–751.
- Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability*, volume 174. W. H. Freeman.
- Gilmore, P. C. and Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120.
- Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422.
- Li, S. (1996). Multi-job cutting stock problem with due dates and release dates. *The Journal of the Operational Research Society*, 47(4):490–510.
- Martin, M., Yanasse, H. H., Santos, M. O., and Morabito, R. (2024). Models for two-dimensional bin packing problems with customer order spread. *Journal of Combinatorial Optimization*, 48:8.
- Oliveira, E. A. S., Wanner, E., de Sá, E. M., and de Souza, S. R. (2025). A local branching-based solution for the multi-period cutting stock problem with tardiness, earliness, and setup costs. *Journal of Heuristics*, 31(1):16.
- Pitombeira-Neto, A. R. and Prata, B. d. A. (2020). A matheuristic algorithm for the one-dimensional cutting stock and scheduling problem with heterogeneous orders. *Top*, 28(1):178–192.
- Polyakovskiy, S. and M’Hallah, R. (2017). A hybrid feasibility constraints-guided search to the two-dimensional bin packing problem with due dates. *European Journal of Operational Research*, 266(3):819–839.
- Reinertsen, H. and Vossen, T. W. M. (2010). The one-dimensional cutting stock problem with due dates. *European Journal of Operational Research*, 201(3):701–711.
- Silva, E., Alvelos, F., and 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(3):699–708.
- Valério de Carvalho, J. M. (1999). Exact solution of bin-packing problems using column generation and branch-and-bound. *Analys of Operations Research*, 86:629–659.