INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

IFORS

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 00 (2019) 1–27
DOI: 10.1111/itor.12648

The multiperiod two-dimensional non-guillotine cutting stock problem with usable leftovers

E. G. Birgin[a], O. C. Romão[a] and D. P. Ronconi[b]

[a] *Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, Brazil*
[b] *Department of Production Engineering, EP-USP, University of São Paulo, Av. Prof. Almeida Prado, 128, Cidade Universitária, 05508-900, São Paulo, Brazil*
E-mail: egbirgin@ime.usp.br [Birgin]; oberlan@ime.usp.br [Romão]; dronconi@usp.br [Ronconi]

Received 24 September 2018; received in revised form 4 January 2019; accepted 16 February 2019

Abstract

A mixed integer linear programing model for the two-dimensional non-guillotine cutting problem with usable leftovers was recently introduced by Andrade *et al.* The problem consists in cutting a set of ordered items using a set of objects of minimum cost and, within the set of solutions of minimum cost, maximizing the value of the usable leftovers. Since the concept of usable leftovers assumes they can potentially be used to attend new arriving orders, the problem is extended to the multiperiod framework in this work. In this way, the decision at each instant does not minimize in a myopic way the cost of the objects required to attend the orders of the current instant; but it aims to minimize the overall cost of the objects up to the considered time horizon. Some variants of the proposed model are analyzed and numerical results are presented.

Keywords: non-guillotine cutting and packing; usable leftovers; MIP models; multiperiod scenario

1. Introduction

Cutting stock problems appear in a wide range of industrial processes where a variety of large pieces of material, such as paper, glass, steel, wood, or fabric, need to be cut in order to produce smaller pieces of ordered sizes and quantities. Aiming to reduce operating costs, several aspects of the cutting process may be taken into account, and returning leftovers to stock so they can be used in future orders is one of them (see, for example, Koch et al. (2009) and Chen et al. (2019), where the usage of leftovers in the wood-processing and plastic-film industries are considered).

In this work, we are concerned with the two-dimensional non-guillotine cutting problem of cutting an heterogeneous set of small rectangular pieces (items) from a set of large rectangular pieces (objects). The problem is said to be two-dimensional because it involves the widths and heights of items and objects, while it is said to be non-guillotine because cuts are not restricted to be guillotine cuts. A multiperiod scenario is considered in which, at each instant p ($0 \leq p \leq P-1$), there are ordered items and available purchasable objects, and items ordered at instant p must be produced within the period $[p, p+1]$. An item ordered at instant p may be produced from a purchased object available at that instant or from a leftover of a previously used object. Thus, we consider that, at each instant, an heterogeneous set of objects is available.

In Andrade et al. (2014), the single-period scenario of the problem described in the paragraph above was tackled. In the single-period scenario, the goal is to minimize the cost of the objects required to produce all ordered items, and, within the set of solutions with minimum cost, to maximize the value of objects' *usable* leftovers. (The formal definition of usable leftover adopted in this work will be given in the next section.) The consideration of usable leftovers assumes their utilization to produce forthcoming orders of items; thus, extending the problem considered in Andrade et al. (2014) to the multiperiod scenario appears as a natural option. In the multiperiod scenario, given a time horizon represented by P instants $0, \dots, P-1$, the goal is to minimize the overall cost of the objects required to produce all items ordered at instants $0, \dots, P-1$, and, within the set of solutions with minimum cost, to maximize the value of usable leftovers available at instant P (i.e., at the end of the considered time horizon). It is very clear that this formulation of the problem is expected to produce better quality solutions than the myopic alternative of solving a single-period problem at each instant. Note that, following Andrade et al. (2014), in the present work there is a clear hierarchy between the two considered objectives—cost must be minimized in the first place and, among solutions of minimum cost, a solution with maximum value of usable leftovers is seeked. This goals' hierarchy fits the tackled problem in the bilevel optimization framework (Dempe, 2002), in opposition to the multiobjective approach (Miettinen, 1998).

Several works were written regarding one-dimensional cutting problems with leftovers (see the pionners' works of Roodman (1986) and Scheithauer (1991), the recent survey by Cherri et al. (2014) and the references therein, and the works of Poldi and Arenales (2010), Cherri et al. (2013), and Tomat and Gradišar (2017)). In Poldi and Arenales (2010), the authors introduce a mixed integer linear programing model, a column generation approach to solve its linear relaxation, and a heuristic rolling horizon approach for rounding off fractional solutions. In Tomat and Gradišar (2017), a multiperiod problem that combines the minimization of the trim-loss and the amount of usable leftovers in stock is considered, and a heuristic method is proposed and tested. In Cherri et al. (2013), to avoid leftovers remaining in stock for a long period of time, it is considered that the leftovers have a priority-in-use compared to standard objects in stock. A heuristic approach is also proposed and tested.

On the other hand, only a few works address two- and three-dimensional cutting problems with leftovers. In Andrade et al. (2014), a mixed integer linear programing model for the two-dimensional non-guillotine cutting stock problem with usable leftovers was introduced. In the considered single-period problem, the goal is to minimize the cost of the objects required to produce a given set of ordered items and, within the set of minimum cost, maximize the value of the usable leftovers. In Andrade et al. (2016), the non-exact two-stage guillotine cutting stock version of the same problem was analyzed. In Viegas et al. (2016), a heuristic cutting decision process for daily tailored orders of a real-life steel retailer is proposed. The considered problem is a three-dimensional cutting and packing problem in which usable leftovers of preceding periods may be used to produce items of the current period. A three-staged two-dimensional cutting stock problem with usable leftover is studied in Chen et al. (2015), where a heuristic beam search approach is developed. Exact and nonexact two- and three-stage two-dimensional cutting stock problems are also considered in Silva et al. (2010). Mixed integer linear programing models, which can be seen as extensions of the model proposed in Dyckhoff (1981) for the one-dimensional cutting stock, are proposed. Introduced models are based on the enumeration of all possible ways of producing an item from an object. Since the production of an item from an object produces the item and also two *residual objects* that can be used to generate other items, this work also considers leftovers. Upper bounds on the number of variables and constraints of the proposed models are given. In Silva et al. (2014), the problem introduced in Silva et al. (2010) is extended to the multiperiod framework and integrated with the lot-sizing problem. In this context, the goal is to minimize a total cost that includes raw material, waste, and storage costs. Mixed integer linear programing models and two heuristic approaches based on the industrial practice are proposed. A multiperiod three-dimensional packing problem is addressed in Alonso et al. (2019), in which the problem of putting products on pallets and then loading the pallets into trucks is considered. Mixed integer linear programing models that include maximum weight constraints as well as stability constraints are presented and tested on real instances related to the everyday distribution activity of a company.

The rest of this paper is organized as follows. Section 2 describes and motivates the multiperiod two-dimensional non-guillotine cutting stock problem with leftovers. Section 3 introduces its mixed integer linear programing formulation. In particular, Section 3.6 introduces a model that minimizes the overall cost of the used objects, whereas Section 3.7 introduces the model that, within the set of solutions with minimum cost, maximizes the value of the leftovers available at the end of the considered time horizon. Section 4 presents illustrative numerical experiments. Conclusions and lines for future research are given in Section 5.

2. The multiperiod two-dimensional non-guillotine cutting stock problem with leftovers

Following Andrade et al. (2014), in this work we consider that object's leftovers are obtained by performing a couple of guillotine precuts on the object that separate the leftovers from the "cutting area" of the object (region from where the items will be cut). As depicted in Fig. 1, there are two possible ways of performing those two cuts: (i) the vertical cut before the horizontal cut or (ii) the horizontal cut before the vertical cut. Given a catalogue of items, we say a leftover is usable if it can fit any item from the catalogue. In this case, the leftover's value is given by its area times the cost per unit of area of the object. Otherwise, the leftover is disposable and has no value at all. This is why the added values of the two leftovers in Fig. 1a may differ from the added values of the two leftovers in Fig. 1b. It is worth noting that this definition of leftovers implies that any part of the cutting area of the object that is not used to produce an item is considered waste. (See Andrade et al. (2014, 2016) for other definitions of leftovers in two-dimensional problems.)

Assume that there are given (i) a set of m available objects $\mathcal{O}_j$ with width W_j, height H_j, and cost c_j per unit of area ($j = 1, \dots, m$), (ii) a set of n ordered items $\mathcal{I}_i$ with width w_i and height h_i ($i = 1, \dots, n$), and (iii) a catalogue composed of d items $\bar{\mathcal{I}}_i$ with width $\bar{w}_i$ and height $\bar{h}_i$ ($i = 1, \dots, d$). Items from the catalogue are used only to determine whether a leftover is usable or not. Consider the problem of cutting all the ordered items from a set of objects of minimum cost. Moreover,

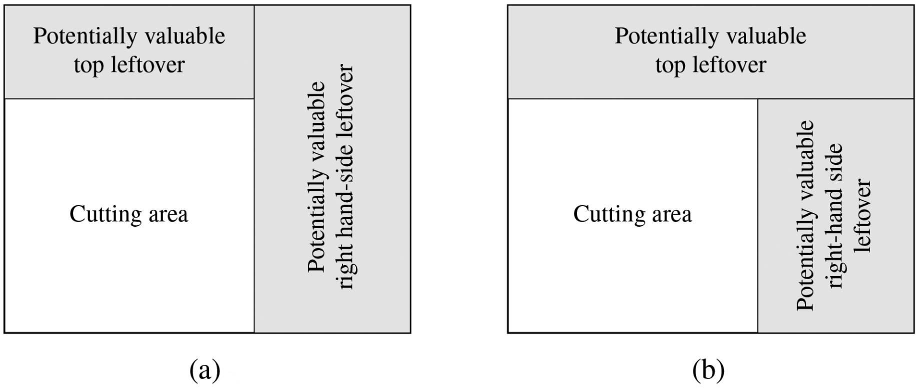


Fig. 1. Graphs (a) and (b) represent the two possible ways of generating leftovers performing a vertical and a horizontal guillotine precut. Leftovers are usable if they can fit any item from a given catalogue of items.

assume that when items are cut from an object, two leftovers can be generated (as described above) and that, among all solutions of minimum cost, we want a solution that maximizes the value of the leftovers. This was one of the problems modeled as a mixed integer linear programing problem in Andrade et al. (2014). Clearly, the idea of considering a leftover usable if it can fit an item from the catalogue assumes that (a) new orders will arrive, (b) new ordered items might be items from the catalogue, and (c) using the leftovers to cut some of the new ordered items might reduce the cost of purchasing new objects. This suggests the existence of an underlying multiperiod framework.

Consider now P instants of time and assume that, at each instant p ($p = 0, \dots, P-1$), there are given (i) a set of m_p available objects $\mathcal{O}_{pj}$ with width W_{pj}, height H_{pj}, and cost c_{pj} per unit of area ($j = 1, \dots, m_p$) and (ii) a set of n_p ordered items $\mathcal{I}_{pi}$ with width w_{pi} and height h_{pi} ($i = 1, \dots, n_p$). A catalogue composed of d items $\bar{\mathcal{I}}_i$ with width $\bar{w}_i$ and height $\bar{h}_i$ ($i = 1, \dots, d$) is also given. At each instant p ($p = 0, \dots, P-1$), we must decide the way of cutting *all* the n_p ordered items. This means that the cut of ordered items cannot be anticipated or delayed. Items may be cut from available objects or from usable leftovers from previous periods. The objective is to minimize the overall cost of the objects required to execute the orders of all instants. Among all solutions of minimum cost, we want a solution that maximizes the value of the leftovers at instant P (the end of the considered time horizon). A leftover is considered valuable if it can fit an item from the catalogue; otherwise, it is disposable and it has no value. We assume this problem is part of a larger scenario within which an agent takes the decision of which orders must be placed at each instant considering the demand and existing constraints related to profit, penalties, stock, labor hours, cash flow, etcetera. This means that the presented problem focuses on the determination of the (non-guillotine) cut patterns that minimize the usage of raw material taking advantage of usable leftovers.

Figure 2 illustrates a small instance of the considered problem. Since, all ordered items must be cut, if the objects available at some instant are not enough to produce all items ordered at that instant, the instance may be infeasible. (The possibility of using leftovers from previous periods exists.) To complete the instance, it must be said that the cost per unit of area of all the objects is one and that the catalogue is composed of the four ordered items. Note that the existence of the catalogue gives some flexibility to the definition of usable leftovers. If, for example, one desires

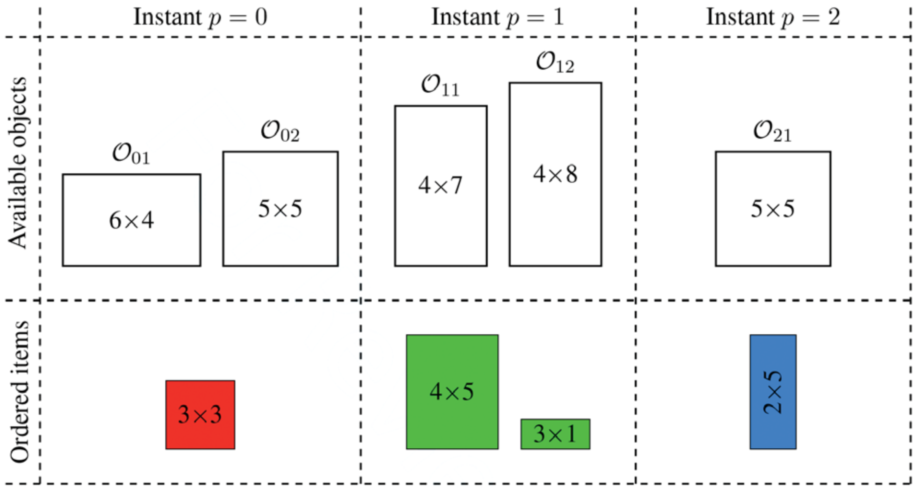


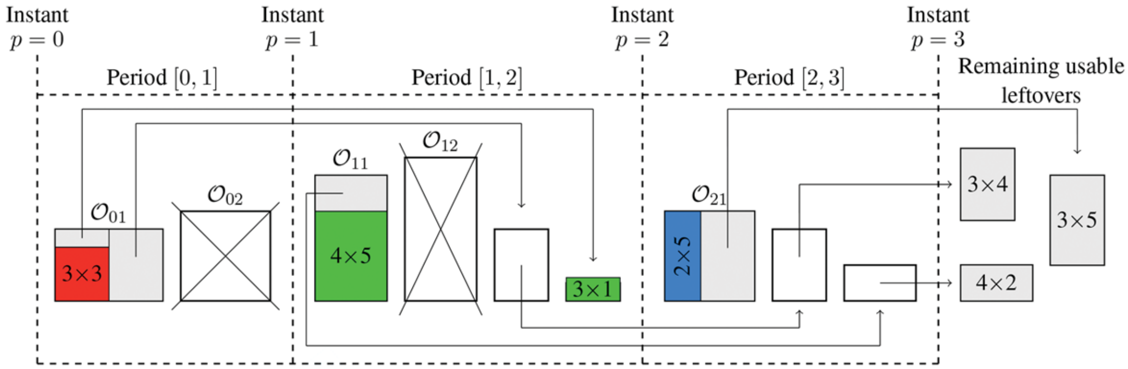
Fig. 2. Illustration a small instance of the considered problem with $P = 3$ periods. The number of available objects at each instant is given by $m_0 = m_1 = 2$ and $m_2 = 1$ and the number of ordered items at each instant is given by $n_0 = n_2 = 1$ and $n_1 = 2$. The cost per unit of area of all the objects is one (i.e., $c_{01} = c_{02} = c_{11} = c_{12} = c_{21} = 1$) and the catalogue with $d = 4$ items is composed of the four ordered items (i.e., $\bar{w}_1 = \bar{h}_1 = 3$, $\bar{w}_2 = 4$, $\bar{h}_2 = 5$, $\bar{w}_3 = 3$, $\bar{h}_3 = 1$, $\bar{w}_4 = 2$, and $\bar{h}_4 = 5$).

leftovers to be usable whenever they have a minimum width $\hat{w}$ and a minimum height $\hat{h}$, then the catalogue may be given by a single item with width $\hat{w}$ and height $\hat{h}$. In this way, leftovers that can fit this item are considered usable and the others are not.

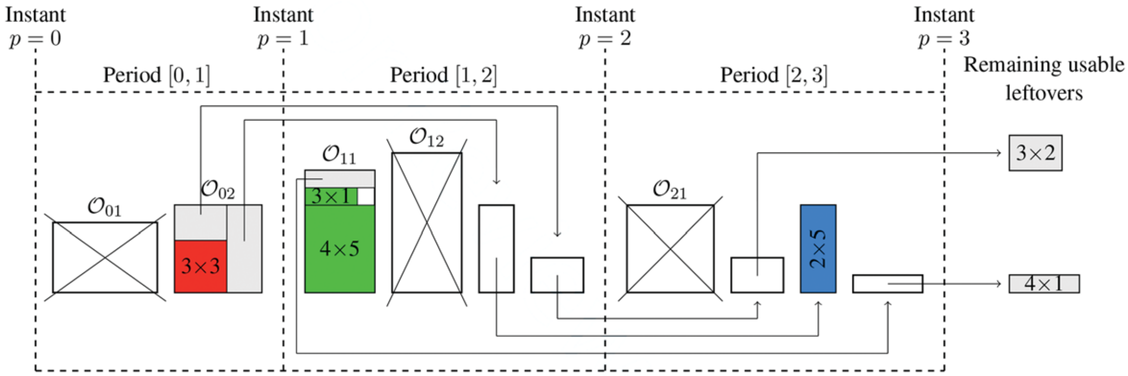
Figure 3 illustrates three different feasible solutions to the instance in Fig. 2. Figure 3a represents a solution that can be found by a myopic approach that proceeds as follows. At each instant, available objects are the objects that can be bought and also the usable leftovers from previous periods (with no cost). The solution to each instant can be found by solving the model introduced in Andrade et al. (2014) that consists in minimizing the cost of the objects required to cut the ordered items and, among solutions with minimum cost, chooses one with maximum value of the usable leftovers. The solution in Fig. 3a uses objects $\mathcal{O}_{01}$, $\mathcal{O}_{11}$, and $\mathcal{O}_{21}$ whose total cost is $24 + 28 + 25 = 77$ and, at instant $p = 3$, has three remaining usable leftovers whose total value is given by $12 + 8 + 15 = 35$. Solutions in Figs. 3b and c use objects $\mathcal{O}_{02}$ and $\mathcal{O}_{11}$ only, whose total cost is $25 + 28 = 53$, implying that the solution in Fig. 3a is not optimal. Note that the smaller total cost of both solutions was obtained by buying a more expensive object at instant $p = 0$. Moreover, the solution in Fig. 3b has usable leftovers at instant $p = 3$ whose total value is $6 + 4 = 10$, while Fig. 3c has usable leftovers at instant $p = 3$ whose total value is $3 + 8 = 11$, being in fact the optimal solution we are looking for.

3. Mixed integer linear programing formulation

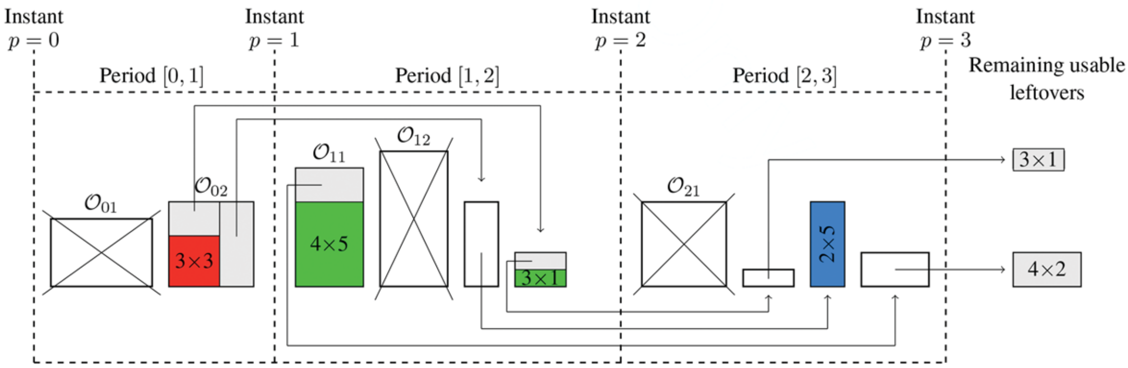
In this section, a mixed integer linear programing formulation for the multiperiod two-dimensional non-guillotine cutting problem with usable leftovers, described in the previous section, is given. The



(a)



(b)



(c)

Fig. 3. Illustration of solutions that, at each period, may cut ordered items from usable leftovers from previous periods. (a) Greedy solution obtained by a myopic method that, *at each instant*, minimizes the cost of the objects required to cut the ordered items of that instant, assuming that usable leftovers from previous periods are free. (b) Solution that minimizes the overall cost of the required objects. (c) Solution with minimum cost of the required objects and, in addition, maximum value of the usable leftovers at instant $p = 3$.

formulation is an extension of the single-period model considered in Andrade et al. (2014) and the novelty relies on the existence, on a given period, of objects that are usable leftovers of previous periods. The dimensions of that objects are not constants but depend on the cutting patterns of the previous periods.

3.1. Instance data

Let P be the number of instants to be considered. Assume that, for each $p = 0, \dots, P-1$, there are given (a) $m_p \geq 0$ and a set of m_p available objects $\mathcal{O}_{pj}$ with width W_{pj}, height H_{pj}, and cost c_{pj} per unit of area ($j = 1, \dots, m_p$) and (b) $n_p \geq 0$ and a set of n_p ordered items $\mathcal{I}_{pi}$ with width w_{pi} and height h_{pi} ($i = 1, \dots, n_p$). A catalogue composed of d items $\bar{\mathcal{I}}_i$ with width $\bar{w}_i$ and height $\bar{h}_i$ ($i = 1, \dots, d$) is also given. As explained in the previous section, each object available at instant p generates two leftovers (that may be usable or not) at instant $p+1$, those two leftovers generate two leftovers *each* at instant $p+2$, and so on. This means that after Δp instants there will be $2^{\Delta p}$ leftovers associated with each object of instant p. This fact would make intractable (to be solved to global optimality) instances with even moderate values of P. Therefore, from the theoretical point of view, it makes sense to add to the problem an integer parameter $\xi \in [0, P]$ that says for how many periods leftovers of an object may be available. If $\xi = 0$, then the problem considers no leftovers at all. If $\xi = P$, then all leftovers will be available up to instant P. Parameter ξ is considered to be the same for all objects of all instants with the only purpose of simplifying the presentation. In practice, each object of each instant might have its own "duration" parameter ξ that would represent the perishability of the raw material it is made of.

3.2. Additional computable constants

By definition, each object generates two leftovers, and leftovers generated in a period $[p, p+1]$ remain available up to period $[p+\xi, p+\xi+1]$; $\xi = 0$ meaning that leftovers are not being considered at all. This means that the number $\bar{m}_p \geq m_p$ of available objects at a given period p, composed of the m_p purchasable objects plus the objects that are leftovers of previous periods, is given by

$$\bar{m}_p = \begin{cases} m_p, & p = 0, \\ m_p + \text{leftovers}(p, \xi), & p = 1, \dots, P-1, \\ \text{leftovers}(p, \xi), & p = P, \end{cases} \tag{1}$$

where

$$\text{leftovers}(p, \xi) = \sum_{\ell=1}^{\min\{p,\xi\}} 2^{\ell} m_{p-\ell}.$$

Note that, since, by definition, there are no purchasable objects at instant P, $\bar{m}_P$ represents the *number of leftovers* available at instant P only.

By definition, the cost (or value) per unit of area of a leftover is the cost per unit of area of the object that originated the leftover. We aim to define $\bar{c}_{pj}$ as the cost per unit of area of each object $\mathcal{O}_{pj}$ ($p = 0, \dots, P$, $j = 1, \dots, \bar{m}_p$). If we name $\mathcal{O}_{p+1,j_1}$ and $\mathcal{O}_{p+1,j_2}$, with $j_1 = m_{p+1} + 2j - 1$ and $j_2 = m_{p+1} + 2j$, the leftovers generated by object $\mathcal{O}_{pj}$ ($p = 0, \dots, P-1$, $j = 1, \dots, \bar{m}_p$), then we have

$$\bar{c}_{p+1,j_1} = \bar{c}_{p+1,j_2} = \bar{c}_{pj} = c_{pj} \text{ for } p = 0, \dots, P-1, j = 1, \dots, m_p.$$

The relevant costs that will be used later in this section are the costs $\bar{c}_{Pj}$ ($j = 1, \dots, \bar{m}_P$) that correspond to the value (per unit of area) of the leftovers available at instant P, that is, at the end of the considered time horizon.

3.3. Assignment of items to objects

Let $v_{pij} \in \{0, 1\}$ ($p = 0, \dots, P-1, i = 1, \dots, n_p, j = 1, \dots, \bar{m}_p$) be such that $v_{pij} = 1$ if item $\mathcal{I}_{pi}$ must be produced from object $\mathcal{O}_{pj}$ and $v_{pij} = 0$, otherwise. The fact that each item must be produced from exactly one object can be modeled with the constraints

$$\sum_{j=1}^{\bar{m}_p} v_{pij} = 1, \;\; p = 0, \dots, P-1, i = 1, \dots, n_p. \tag{2}$$

Let $u_{pj} \in \{0, 1\}$ ($p = 0, \dots, P-1, \;\; j = 1, \dots, \bar{m}_p$) be such that $u_{pj} = 1$ if at least one item is produced from the object $\mathcal{O}_{pj}$ and $u_{pj} = 0$, otherwise. This can be modeled with the constraints

$$u_{pj} \geq v_{pij}, \;\; p = 0, \dots, P-1, j = 1, \dots, \bar{m}_p, i = 1, \dots, n_p, \tag{3}$$

and

$$u_{pj} \leq \sum_{i=1}^{n_p} v_{pij}, \;\; p = 0, \dots, P-1, \;\; j = 1, \dots, \bar{m}_p. \tag{4}$$

3.4. Objects' dimensions

Let $\bar{W}_{pj}$ and $\bar{H}_{pj}$ ($p = 0, \dots, P$, $j = 1, \dots, \bar{m}_p$) be the width and height of object $\mathcal{O}_{pj}$, respectively. Clearly, for every p and $j \leq m_p$, we have that $\bar{W}_{pj} = W_{pj}$ and $\bar{H}_{pj} = H_{pj}$. We now turn our attention to the dimensions $\bar{W}_{pj}$ and $\bar{H}_{pj}$ for $j \in \{m_{p+1}, \dots, \bar{m}_p\}$, that is, dimensions of objects $\mathcal{O}_{pj}$ with $p > 0$ that are leftovers from previous periods. Assuming that the dimensions of all objects of a given instant p are already known (and this is true for $p = 0$), we show how to determine the dimensions of the objects of instant $p + 1$.

Let $p \in \{0, \dots, P-1\}$ be an instant and let t_{pj} and r_{pj} ($j = 1, \dots, \bar{m}_p$), satisfying

$$0 \le t_{pj} \le \bar{H}_{pj} \text{ and } 0 \le r_{pj} \le \bar{W}_{pj}, \;\; j = 1, \dots, \bar{m}_p, \tag{5}$$

be such that $\bar{H}_{pj} - t_{pj}$ and $\bar{W}_{pj} - r_{pj}$ are the height and the width, respectively, of the "cutting area" of object $\mathcal{O}_{pj}$ (see Fig. 1), and let $\eta_{pj} \in \{0, 1\}$ be such that $\eta_{pj} = 1$ if the vertical precut is made in first place (see Fig. 1a) and $\eta_{pj} = 0$ otherwise (see Fig. 1b).

With the help of these three variables (t_{pj}, r_{pj}, and η_{pj}), we are able to determine the dimensions of the two leftovers generated by *the usage or not* of object $\mathcal{O}_{pj}$ that correspond to two available objects at instant $p+1$. As it can be seen in Fig. 1, if the object *is used* and $\eta_{pj} = 1$, then the leftover at the top of the object has width $\bar{W}_{pj} - r_{pj}$ and height t_{pj}, while the leftover on the right-hand side of the object has width r_{pj} and height $\bar{H}_{pj}$. On the other hand, if the object *is used* and $\eta_{pj} = 0$, then the leftover at the top of the object has width $\bar{W}_{pj}$ and height t_{pj}, while the leftover on the right-hand side of the object has width r_{pj} and height $\bar{H}_{pj} - t_{pj}$. When the object *is not used*, we must consider in separate the case in which the object is a purchasable object ($1 \le j \le m_p$) and the case in which the object is a leftover from a previous period ($m_p + 1 \le j \le \bar{m}_p$).

In the case of an unused purchasable object $\mathcal{O}_{pj}$, since an object that is not purchased generates no leftovers, we must define the dimensions of its leftovers as null. Thus, we can model the dimensions of the leftovers of a purchasable object as

$$\begin{aligned}
0 \le \bar{H}_{p+1,j_1} &\le \hat{H} u_{pj}, \\
t_{pj} - (1 - u_{pj})\hat{H} \le \bar{H}_{p+1,j_1} &\le t_{pj} + (1 - u_{pj})\hat{H}, \\
0 \le \bar{W}_{p+1,j_1} &\le \hat{W} u_{pj}, \\
\bar{W}_{pj} - r_{pj} - (1 - \eta_{pj})\hat{W} - (1 - u_{pj})\hat{W} \le \bar{W}_{p+1,j_1} &\le \bar{W}_{pj} - r_{pj} + (1 - \eta_{pj})\hat{W} + (1 - u_{pj})\hat{W}, \\
\bar{W}_{pj} - \eta_{pj}\hat{W} - (1 - u_{pj})\hat{W} \le \bar{W}_{p+1,j_1} &\le \bar{W}_{pj} + \eta_{pj}\hat{W} + (1 - u_{pj})\hat{W}, \\
0 \le \bar{W}_{p+1,j_2} &\le \hat{W} u_{pj}, \\
r_{pj} - (1 - u_{pj})\hat{W} \le \bar{W}_{p+1,j_2} &\le r_{pj} + (1 - u_{pj})\hat{W}, \\
0 \le \bar{H}_{p+1,j_2} &\le \hat{H} u_{pj}, \\
\bar{H}_{pj} - (1 - \eta_{pj})\hat{H} - (1 - u_{pj})\hat{H} \le \bar{H}_{p+1,j_2} &\le \bar{H}_{pj} + (1 - \eta_{pj})\hat{H} + (1 - u_{pj})\hat{H}, \\
\bar{H}_{pj} - t_{pj} - \eta_{pj}\hat{H} - (1 - u_{pj})\hat{H} \le \bar{H}_{p+1,j_2} &\le \bar{H}_{pj} - t_{pj} + \eta_{pj}\hat{H} + (1 - u_{pj})\hat{H},
\end{aligned} \tag{6}$$

for $j = 1, \dots, m_p$, where $j_1 = m_{p+1} + 2j - 1$, $j_2 = m_{p+1} + 2j$, and the constants $\hat{W}$ and $\hat{H}$ are given by $\hat{W} = \max\{W_{pj} \mid p = 0, \dots, P-1, j = 1, \dots, m_p\}$ and $\hat{H} = \max\{H_{pj} \mid p = 0, \dots, P-1, j = 1, \dots, m_p\}$.

In the case of an unused object $\mathcal{O}_{pj}$ that is a leftover from a previous period, we must have an object $\mathcal{O}_{p+1,j_1}$ identical to $\mathcal{O}_{pj}$ and an object $\mathcal{O}_{p+1,j_2}$ with null dimensions, or the analogous situation in which objects $\mathcal{O}_{p+1,j_1}$ and $\mathcal{O}_{p+1,j_2}$ change their places. If we arbitrarily consider the first case, we can model the dimensions of the leftovers of an object that is a leftover from a previous period as

$$\begin{aligned}
\bar{H}_{pj} - \hat{H}u_{pj} \le \bar{H}_{p+1,j_1} &\le \bar{H}_{pj} + \hat{H}u_{pj},\\
t_{pj} - (1-u_{pj})\hat{H} \le \bar{H}_{p+1,j_1} &\le t_{pj} + (1-u_{pj})\hat{H},\\
\bar{W}_{pj} - \hat{W}u_{pj} \le \bar{W}_{p+1,j_1} &\le \bar{W}_{pj} + \hat{W}u_{pj},\\
\bar{W}_{pj} - r_{pj} - (1-\eta_{pj})\hat{W} - (1-u_{pj})\hat{W} \le \bar{W}_{p+1,j_1} &\le \bar{W}_{pj} - r_{pj} + (1-\eta_{pj})\hat{W} + (1-u_{pj})\hat{W},\\
\bar{W}_{pj} - \eta_{pj}\hat{W} - (1-u_{pj})\hat{W} \le \bar{W}_{p+1,j_1} &\le \bar{W}_{pj} + \eta_{pj}\hat{W} + (1-u_{pj})\hat{W},\\
0 \le \bar{W}_{p+1,j_2} &\le \hat{W}u_{pj},\\
r_{pj} - (1-u_{pj})\hat{W} \le \bar{W}_{p+1,j_2} &\le r_{pj} + (1-u_{pj})\hat{W},\\
0 \le \bar{H}_{p+1,j_2} &\le \hat{H}u_{pj},\\
\bar{H}_{pj} - (1-\eta_{pj})\hat{H} - (1-u_{pj})\hat{H} \le \bar{H}_{p+1,j_2} &\le \bar{H}_{pj} + (1-\eta_{pj})\hat{H} + (1-u_{pj})\hat{H},\\
\bar{H}_{pj} - t_{pj} - \eta_{pj}\hat{H} - (1-u_{pj})\hat{H} \le \bar{H}_{p+1,j_2} &\le \bar{H}_{pj} - t_{pj} + \eta_{pj}\hat{H} + (1-u_{pj})\hat{H},
\end{aligned} \tag{7}$$

for $j = m_p + 1, \dots, \bar{m}_p$. Note that (6) and (7) differ only in the constraints that apply to $\bar{W}_{p+1,j_1}$ and $\bar{H}_{p+1,j_1}$ when $u_{pj} = 0$. While (6) says that in this case we must have $\bar{W}_{p+1,j_1} = \bar{H}_{p+1,j_1} = 0$, (7) says that it must hold $\bar{W}_{p+1,j_1} = \bar{W}_{pj}$ and $\bar{H}_{p+1,j_1} = \bar{H}_{pj}$.

A technicality is missing and, therefore, there is some abuse of notation in the description of constraints (6) and (7). In both constraints, it is assumed that every object generates two leftovers (as it does in fact when $\xi = P$). Thus, it is written that constraint (6) applies to all $j = 1, \dots, m_p$, constraints (7) applies to all $j = m_p + 1, \dots, \bar{m}_p$, and we define $j_1 = m_{p+1} + 2j - 1$ and $j_2 = m_{p+1} + 2j$. In practice, every object $\mathcal{O}_{pj}$ has an associated "shelf life." A purchasable object has shelf life ξ, while an object that is a leftovers has a shelf life that is one less than the shelf life of the object that generated the leftover. Then, only objects with a strictly positive shelf life generate leftovers, and the leftovers must be numbered accordingly. For example, if, at a period p, $\mathcal{O}_{p,j_a}, \mathcal{O}_{p,j_b}, \dots$ (with $j_a \le j_b \le \cdots$) are the objects that generate leftovers (i.e., the objects with a strictly positive shelf life), then the two leftovers of $\mathcal{O}_{p,j_a}$ should be numbered $m_{p+1} + 1$ and $m_{p+1} + 2$, whereas the two leftovers of $\mathcal{O}_{p,j_b}$ should be numbered $m_{p+1} + 3$ and $m_{p+1} + 4$. In any case, note that the number of leftovers at every period p, given by $\bar{m}_p - m_p$, where $\bar{m}_p$ is defined in (1), is fixed and it depends only on the instance data and the additional computable constants described in Sections 3.1 and 3.2.

3.5. Avoiding overlapping and fitting items within objects' "cutting area"

We now consider the positioning constraints that avoid overlapping of items produced from the same object and the constraints that fit the items within the cutting area of the objects. For this, let (x_{pi}, y_{pi}) be the Cartesian coordinates of the center of item $\mathcal{I}_{pi}$ ($p = 0, \dots, P-1$, $i = 1, \dots, n_p$). The fitting constraints given by

$$
\begin{aligned}
&0 \le x_{pi} - \frac{1}{2}w_{pi} \text{ and } x_{pi} + \frac{1}{2}w_{pi} \le \bar{W}_{pj} - r_{pj} + (1 - v_{pij})\hat{W},\\
&0 \le y_{pi} - \frac{1}{2}h_{pi} \text{ and } y_{pi} + \frac{1}{2}h_{pi} \le \bar{H}_{pj} - t_{pj} + (1 - v_{pij})\hat{H},
\end{aligned} \tag{8}
$$

for $p = 0, \ldots, P-1$, $i = 1, \ldots, n_p$, $j = 1, \ldots, \bar{m}_p$, say that item $\mathcal{I}_{pi}$ must be placed within the cutting area of the object $\mathcal{O}_{pj}$ to which it was assigned. Note that we have assumed, without loss of generality, that the bottom-left corner of all objects corresponds to the origin of the Cartesian plane.

Let $\mathcal{I}_{pi}$ and $\mathcal{I}_{pi'}$ with $i' > i$ be two items that are being assigned to the same object $\mathcal{O}_{pj}$, that is, $v_{pij} = v_{pi'j} = 1$. The non-overlapping constraints must say that $|x_{pi} - x_{pi'}| \ge \frac{1}{2}(w_{pi} + w_{pi'})$ or $|y_{pi} - y_{pi'}| \ge \frac{1}{2}(h_{pi} + h_{pi'})$. Using that $|a| \ge b$ is the same that $a \ge b$ or $-a \ge b$, these disjunction can be written using their big-M formulation as

$$
\begin{aligned}
x_{pi} - x_{pi'} &\ge \frac{1}{2}(w_{pi} + w_{pi'}) - \hat{W}\big[(1 - v_{pij}) + (1 - v_{pi'j}) + \pi_{pii'} + \tau_{pii'}\big],\\
-x_{pi} + x_{pi'} &\ge \frac{1}{2}(w_{pi} + w_{pi'}) - \hat{W}\big[(1 - v_{pij}) + (1 - v_{pi'j}) + \pi_{pii'} + (1 - \tau_{pii'})\big],\\
y_{pi} - y_{pi'} &\ge \frac{1}{2}(h_{pi} + h_{pi'}) - \hat{H}\big[(1 - v_{pij}) + (1 - v_{pi'j}) + (1 - \pi_{pii'}) + \tau_{pii'}\big],\\
-y_{pi} + y_{pi'} &\ge \frac{1}{2}(h_{pi} + h_{pi'}) - \hat{H}\big[(1 - v_{pij}) + (1 - v_{pi'j}) + (1 - \pi_{pii'}) + (1 - \tau_{pii'})\big],
\end{aligned} \tag{9}
$$

for $p = 0, \ldots, P-1$, $j = 1, \ldots, \bar{m}_p$, $i = 1, \ldots, n_p$, $i' = i+1, \ldots, n_p$, where $\pi_{pii'}$ and $\tau_{pii'} \in \{0, 1\}$ for $p = 0, \ldots, P-1$, $i = 1, \ldots, n_p$, and $i' = i+1, \ldots, n_p$ are auxiliary variables. When two items ordered at the same instant are identical (i.e., they have identical dimensions), additional constraints (introduced in Andrade and Birgin, 2013, §3) may be added in order to avoid symmetric solutions in which both items interchange their places (see also Andrade et al., 2014, p. 1651).

3.6. *Minimizing the cost of the used objects*

Up to this point, we have all the elements to build up the mixed integer linear programing formulation of the problem of minimizing the overall cost of the objects required to satisfy the orders of all instants making use of leftovers. Variables of the problem are $v_{pij} \in \{0, 1\}$ ($p = 0, \ldots, P-1$, $j = 1, \ldots, \bar{m}_p$, $i = 1, \ldots, n_p$), $u_{pj} \in \{0, 1\}$, ($p = 0, \ldots, P-1$, $j = 1, \ldots, m_p$), $\eta_{pj} \in \{0, 1\}$, $\bar{W}_{pj}$, $\bar{H}_{pj}$, t_{pj}, $r_{pj} \in \mathbb{R}$ ($p = 0, \ldots, P-1$, $j = 1, \ldots, \bar{m}_p$), and $\pi_{pii'}$, $\tau_{pii'} \in \{0, 1\}$ ($p = 0, \ldots, P-1$, $i = 1, \ldots, n_p$, $i' = i+1, \ldots, n_p$). Since the cost of the used objects is given by

$$
\sum_{p=0}^{P-1} \sum_{j=1}^{m_p} c_{pj} W_{pj} H_{pj} u_{pj}, \tag{10}
$$

the problem is given by minimizing (10) subject to the constraints (2), (3), and (4) that assign items to objects; the constraints (5), (6), and (7) that determine the dimensions of the leftovers; and the constraints (8) and (9) that avoid overlapping between the items and fit the items within the cutting area of the objects, respectively.

3.7. Maximizing the value of usable leftovers at the end of the time horizon

If we consider the instance depicted in Fig. 2, solutions illustrated in Figs. 3b and 3c are both optimal solutions to the model introduced in Section 3.6. This is because, although leftovers are being used to reduce the cost of the required objects, the value of the usable leftovers available at instant P are *not* being considered in the model to determine that, in fact, the solution in Fig. 3c is preferred. Thus, we now need to model that, by definition, *usable* leftovers are the ones that can fit at least an item from the catalogue, while the other ones are disposable, and that, also by definition, the value of an usable leftover is given by its area times the cost per unit of area of the object that generated the leftover. Then, the value of the usable leftovers available at instant P must be incorporated into the model as a tie break to differentiate solutions with minimum cost of the used objects.

All objects available at instant P are, by definition, leftovers of previous periods, since there are no purchasable objects at this (last) instant. Moreover, they are exactly $\bar{m}_P$ objects and they have width $\bar{W}_{Pj}$ and height $\bar{H}_{Pj}$ for $j = 1, \dots, \bar{m}_P$. Each object $\mathcal{O}_{Pj}$ that can fit an item from the catalogue has value $\bar{c}_{Pj}\bar{W}_{Pj}\bar{H}_{Pj}$, while the others have no value and are disposable. Assume we are able to introduce variables γ_j ($j = 1, \dots, \bar{m}_P$) and additional constraints that impose that $\gamma_j = \bar{W}_{Pj}\bar{H}_{Pj}$ if there exists an item $\mathcal{I}_i$ ($i = 1, \dots, d$) from the catalogue such that $\bar{w}_i \le \bar{W}_{Pj}$ and $\bar{h}_i \le \bar{H}_{Pj}$, and $\gamma_j = 0$ otherwise. In this case, the value of the usable leftovers at instant P is given by

$$\sum_{j=1}^{\bar{m}_P} \bar{c}_{Pj}\gamma_j.$$

Since leftovers come from objects, the value of all leftovers is strictly smaller than the cost of the objects they come from. Thus,

$$\sum_{j=1}^{\bar{m}_P} \bar{c}_{Pj}\gamma_j \le \sum_{j=1}^{\bar{m}_P} \bar{c}_{Pj}\bar{W}_{Pj}\bar{H}_{Pj} < \sum_{p=0}^{P-1}\sum_{j=1}^{m_p} c_{pj}W_{pj}H_{pj}u_{pj} \le \sum_{p=0}^{P-1}\sum_{j=1}^{m_p} c_{pj}W_{pj}H_{pj}.$$

If we assume that c_{pj}, W_{pj}, and H_{pj} ($p = 0, \dots, P-1$, $j = 1, \dots, m_p$) are integer numbers, then minimizing

$$\sum_{p=0}^{P-1}\sum_{j=1}^{m_p} c_{pj}W_{pj}H_{pj}u_{pj} - \left(\frac{1}{\sum_{p=0}^{P-1}\sum_{j=1}^{m_p} c_{pj}W_{pj}H_{pj}} \right) \sum_{j=1}^{\bar{m}_P} \bar{c}_{Pj}\gamma_j,$$

or, equivalently,

$$\left(\sum_{p=0}^{P-1}\sum_{j=1}^{m_p} c_{pj}W_{pj}H_{pj}\right)\left(\sum_{p=0}^{P-1}\sum_{j=1}^{m_p} c_{pj}W_{pj}H_{pj}u_{pj}\right) - \sum_{j=1}^{\bar{m}_P}\bar{c}_{Pj}\gamma_j, \tag{11}$$

has the effect of minimizing the overall cost of the used objects and, within the set of solutions with minimum cost, maximizing the value of the usable leftovers available at instants P.

3.7.1. Modeling the area of the usable leftovers

It remains to describe the constraints that produce the desired effect on the variables γ_j ($j = 1,\dots,\bar{m}_P$), that is,

$$\gamma_j = \begin{cases} \bar{W}_{Pj}\bar{H}_{Pj}, & \text{if there exists } i \in \{1,\dots,d\} \text{ such that } \bar{w}_i \le \bar{W}_{Pj} \text{ and } \bar{h}_i \le \bar{H}_{Pj}, \\ 0, & \text{otherwise.} \end{cases}$$

If, in addition to the integrality of c_{pj}, W_{pj}, and H_{pj} ($p = 0,\dots,P-1$, $j = 1,\dots,m_p$), we also assume that all ordered items have integer dimensions, that is, that w_{pi} and h_{pi} ($p = 0,\dots,P-1$, $j = 1,\dots,m_p$) are all integers, then we have that there are optimal solutions for which $\bar{W}_{Pj}$ and $\bar{H}_{Pj}$ ($j = 1,\dots,\bar{m}_P$) are all integer as well (see, e.g., Birgin et al., 2010, 2012). Therefore, we can express $\bar{W}_{Pj}$ as

$$\bar{W}_{Pj} = \sum_{\ell=1}^{L} 2^{\ell-1}\theta_{j\ell}, \tag{12}$$

where $L = \lfloor \log_2(\hat{W})\rfloor + 1$ and $\theta_{j\ell} \in \{0,1\}$ ($j = 1,\dots,\bar{m}_P$, $\ell = 1,\dots,L$), that is, $\theta_{jL}\theta_{j,L-1}\cdots\theta_{j1}$ being the binary representation of $\bar{W}_{Pj}$ and L being an upper bound on the number of required bits. With this, we have that, for $j = 1,\dots,\bar{m}_P$,

$$\bar{W}_{Pj}\bar{H}_{Pj} = \sum_{\ell=1}^{L} 2^{\ell-1}\bar{H}_{Pj}\theta_{j\ell},$$

thus representing the product of two integers by the sum of products of an integer and a binary variable (see, e.g., Harjunkoski et al., 1997; Yanasse and Morabito, 2006). The value of each of these products coincides with the value of the integer if the binary variable is one and zero otherwise. Introducing (continuous) variables $\omega_{j\ell}$ ($j = 1,\dots,\bar{m}_P$, $\ell = 1,\dots,L$), this products can be modeled as

$$0 \le \omega_{j\ell} \le \bar{H}_{Pj} \text{ and } \bar{H}_{Pj} - (1-\theta_{j\ell})\hat{H} \le \omega_{j\ell} \le \theta_{j\ell}\hat{H} \text{ for } j = 1,\dots,\bar{m}_P,\ \ell = 1,\dots,L. \tag{13}$$

Up to now, we have that, with the variables $\theta_{j\ell} \in \{0, 1\}$ and $\omega_{j\ell}$ ($j = 1, \dots, \bar{m}_P$, $\ell = 1, \dots, L$) and the constraints (13), each product $\bar{W}_{Pj}\bar{H}_{Pj}$ ($j = 1, \dots, \bar{m}_P$) is given by

$$\sum_{\ell=1}^{L} 2^{\ell-1}\omega_{j\ell}.$$

Now consider variables $\zeta_{ji} \in \{0, 1\}$ ($j = 1, \dots \bar{m}_P$, $i = 1, \dots, d$). The idea is that $\zeta_{ji} = 0$ if item $\mathcal{I}_i$ from the catalogue does not fit within object $\mathcal{O}_{Pj}$, that is, if $\bar{w}_i > \bar{W}_{Pj}$ or $\bar{h}_i > \bar{H}_{Pj}$. We model this with the constraints

$$\bar{w}_i \le \bar{W}_{Pj} + \hat{W}(1 - \zeta_{ji}) \text{ and } \bar{h}_i \le \bar{H}_{Pj} + \hat{H}(1 - \zeta_{ji}) \text{ for } j = 1, \dots, \bar{m}_P,\ i = 1, \dots, d. \tag{14}$$

Now, constraints

$$0 \le \gamma_j \le \sum_{\ell=1}^{L} 2^{\ell-1}\omega_{j\ell} \text{ and } \gamma_j \le \left(\sum_{i=1}^{d} \zeta_{ji}\right)\hat{W}\hat{H} \text{ for } j = 1, \dots, \bar{m}_P \tag{15}$$

say that the value of γ_j may vary between zero and the area of object $\mathcal{O}_{Pj}$, and that $\gamma_j = 0$, if no item from the catalogue fits within object $\mathcal{O}_{Pj}$. Since each γ_j appears with a negative coefficient in the objective function being minimized, in any solution we will have γ_j equal to its maximum possible value as desired.

3.7.2. The full model

Summing up, we now have all the ingredients to build up the complete mixed integer linear programing formulation of the problem of minimizing the overall cost of the objects required to satisfy the demand of instants from 0 to $P - 1$ making use of leftovers, and, among all solutions with minimum cost, maximizing the value of the usable leftovers at instant P.

Variables of the problem are $v_{pij} \in \{0, 1\}$ ($p = 0, \dots, P - 1$, $j = 1, \dots, \bar{m}_p$, $i = 1, \dots, n_p$), $u_{pj} \in \{0, 1\}$ ($p = 0, \dots, P$, $j = 1, \dots, m_p$), $\eta_{pj} \in \{0, 1\}$, t_{pj}, $r_{pj} \in \mathbb{R}$ ($p = 0, \dots, P - 1$, $j = 1, \dots, \bar{m}_p$), $\bar{W}_{pj}$, $\bar{H}_{pj}$ ($p = 0, \dots, P$, $j = 1, \dots, \bar{m}_p$), $\pi_{pii'}$, $\tau_{pii'} \in \{0, 1\}$ ($p = 0, \dots, P - 1$, $i = 1, \dots, n_p$, $i' = i + 1, \dots, n_p$), γ_j, $\theta_{j\ell} \in \{0, 1\}$, $\omega_{j\ell}$ ($j = 1, \dots, \bar{m}_P$, $\ell = 1, \dots, L$), and $\zeta_{ji} \in \{0, 1\}$ ($j = 1, \dots, \bar{m}_P$, $i = 1, \dots, d$).

The problem is given by minimize (11) subject to the constraints (2), (3), and (4) that assign items to objects; the constraints (5), (6), and (7) that determine the dimensions of the leftovers; the constraints (8) and (9) that avoid overlapping between the items and fit the items within the cutting area of the objects, respectively; and the constraints (13), (14), and (15) that model the value of the usable leftovers at instant P. (Note that, due to (12), or (12) for $j = 1, \dots, \bar{m}_P$ is added to the model or variables $\bar{W}_{Pj}$ ($j = 1, \dots, \bar{m}_P$) are eliminated as variables and all their occurrences are replaced by the right-hand side of (12). In the numerical experiments included in the next section, we arbitrarily opted by including (12) for $j = 1, \dots, \bar{m}_P$ as constraints of the model.)

4. Illustrative numerical examples

In this section, we present numerical experiments with the proposed model. The goal of the numerical experiments is to analyze the influence of considering leftovers in the overall cost of the purchased objects, so each considered instance will be solved varying $\xi \in \{0, 1, \ldots, P\}$. Recall that $\xi = 0$ means that leftovers are not considered at all, while $\xi = P$ means that leftovers generated at any period are available up to the end of the considered time horizon. Twenty-five small-sized instances with up to four periods will be solved with an exact commercial solver. Table 1 describes the instances. In the table, for each instance, P is the number of periods. For each instant $p = 0, 1, \ldots, P-1$, m_p is the number of purchasable objects and n_p is the number of ordered items. Notation $a(b \times c)[s]$ means that there are a objects or items with width b and height c, and, in the case of objects, that the cost per unit of area is s. When a is omitted, it means that there is a single copy of the described object or item, and, when s is omitted, it means that the cost per unit of area is 1. In the last column, d is the number of items in the catalogue. Items in the catalogue are the ones whose dimensions are underlined in the table. Table 2 displays the number of binary and continuous variables and the number of constraints of each one of 25 considered instances varying $\xi \in \{0, 1, \ldots, P\}$. From the table, it is easy to see how these figures grow as a function of ξ.

The model was implemented in C/C++ using the ILOG Concert Technology and compiled with g++ from gcc version 5.4.0 (GNU compiler collection) with the -O3 option enable. Numerical experiments were conducted using a machine with Intel Xeon Processor X5650, 8 GB of RAM memory, and Ubuntu 16.04 operating system. Instances were solved using IBM ILOG CPLEX 12.8.0. By default, a solution is reported as optimal by the solver when

$$\text{absolute gap} = \text{best feasible solution} - \text{best lower bound} \leq \varepsilon_{\text{abs}},$$

or

$$\text{relative gap} = \frac{|\text{ best feasible solution} - \text{best lower bound }|}{10^{-10} + |\text{ best feasible solution }|} \leq \varepsilon_{\text{rel}},$$

with $\varepsilon_{\text{abs}} = 10^{-6}$ and $\varepsilon_{\text{rel}} = 10^{-4}$, where "best feasible solution" means the smallest value of the objective function related to a feasible solution generated by the method. The objective function (11) has the particular property of assuming relatively large integer values at feasible points. Hence, a stopping criterion based on a relative error less than or equal to $\varepsilon_{\text{rel}} = 10^{-4}$ may have the undesired effect of stopping the method prematurely. On the other hand, due to the integrality of the objective function values, an absolute error strictly smaller than 1 is enough to prove the optimality of the incumbent solution. Therefore, in the numerical experiments, we considered $\varepsilon_{\text{abs}} = 1 - 10^{-6}$ and $\varepsilon_{\text{rel}} = 0$. In addition, `NodeFileInd` and `WorkMem` parameters were set to 3 and 6,000, respectively, so the branch and bound tree is partially transferred to disk if memory is exhausted. All other parameters of the solver were used with their default values.

Tables 3–5 describe the solutions found for the 25 considered instances for varying values of $\xi \in \{0, 1\}$, $\xi \in \{2, 3\}$, and $\xi = 4$, respectively. In the tables, "objective function optimal value" is the value of the objective function (11) at the solution reported as optimal by the solver. "Objects costs" and "leftovers value" correspond to the cost of the purchased objects and the value of the leftovers at instant P, respectively, and they are extracted from the optimal value according to (11). A CPU time limit of two hours was imposed to the solver. When this time limit is reached, as is the case

Table 1
Description of the considered set of instances

			Objects		Items		
#Inst.	P	p	m_p	$W \times H$	n_p	$w \times h$	d
1	3	0	3	21 × 17, 19 × 19, 24 × 13	2	10 × 11, 9 × 11	1
		1	1	10 × 16	3	7 × 6, 7 × 5, 7 × 4	
		2	1	10 × 12	2	2($\underline{6 \times 3}$)	
2	4	0	2	14 × 8, 16 × 6	3	3 × 7, 6 × 8, 4 × 8	2
		1	1	15 × 10	3	5 × 3, 2(2 × 5)	
		2	1	20 × 15	2	5 × 3, $\underline{3 \times 2}$	
		3	1	15 × 10	2	2($\underline{2 \times 3}$)	
3	4	0	2	15 × 6, 15 × 5	3	2($\underline{1 \times 6}$), 10 × 6	2
		1	1	12 × 7	1	3 × 5	
		2	1	20 × 10	2	5 × 3, 3 × 2	
		3	1	20 × 8	6	2(2 × 3), 10 × 1, $\underline{2 \times 2}$, 2(5 × 2)	
4	4	0	2	13 × 8, 12 × 6	5	1 × 5, 2 × 5, 1 × 4, 1 × 3, 3 × 2	2
		1	3	10 × 8, 12 × 10, 15 × 10	3	3 × 7, 2 × 3, 2 × 4	
		2	1	8 × 4	2	10 × 1, $\underline{1 \times 3}$	
		3	0		3	$\underline{3 \times 1}$, 3 × 3, 4 × 4	
5	4	0	2	10 × 4, 13 × 8	4	2(1 × 5), 2 × 5, 3 × 5	1
		1	2	10 × 9, 12 × 9	2	5 × 3, 6 × 3	
		2	3	10 × 10, 2(12 × 9)	3	5 × 3, 6 × 2, 3 × 3	
		3	0		3	$\underline{1 \times 2}$, 5 × 4, 4 × 2	
6	4	0	2	22 × 17, 14 × 30	5	3($\underline{2 \times 11}$), 2(5 × 5)	4
		1	2	17 × 29, 24 × 10	2	2(4 × 10)	
		2	2	18 × 19, 26 × 22	3	3(5 × 4)	
		3	3	24 × 12, 15 × 18, 17 × 13	8	4($\underline{3 \times 3}$), $\underline{4 \times 2}$, 2($\underline{7 \times 1}$), 11 × 1	
7	4	0	2	(10 × 12)[2], 12 × 10	3	5 × 4, 8 × 2, $\underline{2 \times 2}$	1
		1	1	17 × 15	1	3 × 7	
		2	1	17 × 15	1	8 × 4	
		3	1	17 × 15	1	4 × 9	
8	4	0	2	10 × 12, (12 × 10)[2]	3	5 × 4, 8 × 2, $\underline{2 \times 2}$	1
		1	1	17 × 15	1	3 × 7	
		2	1	17 × 15	1	8 × 4	
		3	1	17 × 15	1	4 × 9	
9	4	0	3	30 × 20, 2(10 × 10)[3]	6	3 × 7, 8 × 2, 10 × 1, 5 × 4, 2 × 9, $\underline{2 \times 2}$	2
		1	3	(30 × 20)[3], 2(10 × 10)[3]	6	5 × 3, 9 × 3, 6 × 1, 3 × 8, 4 × 1, 7 × 3	
		2	0		4	3 × 2, 7 × 2, 4 × 5, $\underline{4 \times 1}$	
		3	0		4	8 × 4, 4 × 2, 3 × 7, $\underline{6 \times 2}$	
10	4	0	2	14 × 21, 19 × 19	7	2(11 × 3), 3(2 × 11), 2(5 × 5)	1
		1	1	27 × 23	9	9 × 7, 4(9 × 6), 2(5 × 3), 2(5 × 4)	
		2	1	20 × 15	9	5(3 × 2), 4(3 × 1)	
		3	1	17 × 17	7	4(3 × 4), 3($\underline{2 \times 1}$)	

Continued

Table 1
Continued

			Objects		Items		
#Inst.	P	p	m_p	$W \times H$	n_p	$w \times h$	d
11	4	0	2	30 × 10, 23 × 16	1	6 × 6	2
		1	1	28 × 12	3	2 × 5, 2(4 × 1)	
		2	2	22 × 11, 26 × 23	3	2(9 × 3), 6 × 6	
		3	1	17 × 29	3	2(4 × 3), 7 × 2	
12	4	0	2	37 × 20, 22 × 24	2	2(11 × 6)	1
		1	1	21 × 23	1	6 × 6	
		2	1	36 × 30	2	2(13 × 5)	
		3	2	13 × 18, 10 × 17	2	4 × 5, 4 × 2	
13	4	0	2	25 × 34, 36 × 14	2	2(6 × 6)	2
		1	2	23 × 18, 33 × 33	1	6 × 3	
		2	1	17 × 26	1	1 × 6	
		3	2	38 × 23, 30 × 36	1	4 × 10	
14	4	0	1	40 × 33	4	2(3 × 12), 2(15 × 10)	1
		1	1	26 × 36	4	2(3 × 4), 2(10 × 9)	
		2	1	13 × 19	4	2(5 × 3), 2(2 × 3)	
		3	1	32 × 19	2	2(8 × 6)	
15	4	0	2	10 × 24, 26 × 38	2	2(11 × 13)	2
		1	1	25 × 23	2	2(6 × 2)	
		2	1	36 × 36	4	2(3 × 4), 2(6 × 13)	
		3	1	39 × 25	4	2(2 × 4), 2(14 × 3)	
16	4	0	3	20 × 38, 2(11 × 17)	4	2(2 × 4), 2(6 × 16)	3
		1	1	33 × 21	2	2(8 × 9)	
		2	1	12 × 22	2	2(4 × 2)	
		3	1	30 × 14	2	2(5 × 1)	
17	4	0	1	15 × 39	3	2(6 × 2), 5 × 9	2
		1	1	19 × 13	4	2(7 × 2), 2(5 × 6)	
		2	1	20 × 40	2	2(3 × 4)	
		3	2	38 × 40, 22 × 26	3	2(4 × 13), 4 × 8	
18	4	0	1	22 × 38	1	2 × 11	1
		1	3	2(22 × 12), 33 × 17	4	2(14 × 5), 2(12 × 7)	
		2	2	12 × 13, 23 × 11	2	2(7 × 5)	
		3	2	10 × 23, 14 × 20	3	2(1 × 2), 4 × 10	
19	4	0	2	14 × 14, 39 × 11	2	11 × 6, 8 × 5	2
		1	1	15 × 23	3	2(6 × 10), 2 × 10	
		2	1	39 × 14	3	2(5 × 5), 7 × 2	
		3	1	36 × 11	2	3 × 1, 3 × 2	
20	4	0	1	27 × 24	3	4 × 6, 2(10 × 2)	4
		1	2	35 × 27, 27 × 11	1	14 × 5	
		2	2	23 × 30, 17 × 13	3	2(6 × 8), 5 × 5	
		3	1	24 × 34	2	2(3 × 7)	

Continued

Table 1
Continued

			Objects		Items		
#Inst.	P	p	m_p	$W \times H$	n_p	$w \times h$	d
21	4	0	3	10 × 17, 26 × 15, 12 × 11	1	2 × 1	1
		1	1	23 × 20	1	10 × 3	
		2	3	11 × 16, 22 × 15, 28 × 30	1	3 × 10	
		3	1	30 × 28	2	2(8 × 2)	
22	4	0	2	16 × 24, 20 × 10	4	5 × 9, 8 × 6, 2(2 × 4)	1
		1	1	11 × 13	1	2 × 5	
		2	3	22 × 17, 13 × 11, 29 × 29	1	3 × 7	
		3	2	30 × 23, 18 × 23	2	2(4 × 8)	
23	4	0	3	16 × 12, 12 × 10, 19 × 25	6	2(4 × 5), 2(1 × 10), 2(4 × 3)	2
		1	3	18 × 20, 25 × 13, 21 × 16	2	2(2 × 5)	
		2	2	12 × 24, 14 × 16	5	2(2 × 2), 5 × 9, 2(6 × 2)	
		3	1	14 × 27	4	3 × 6, 2(4 × 6), 1 × 4	
24	4	0	1	21 × 21	5	4 × 2, 2(3 × 9), 2(8 × 3)	3
		1	2	19 × 30, 23 × 12	3	2 × 6, 8 × 5, 5 × 4	
		2	2	21 × 28, 24 × 11	1	10 × 2	
		3	1	29 × 16	2	2(3 × 5)	
25	4	0	3	22 × 28, 30 × 25, 19 × 22	2	2(6 × 5)	2
		1	2	22 × 22, 12 × 22	4	2(4 × 8), 2(2 × 3)	
		2	1	22 × 11	4	3 × 3, 3 × 1, 2(8 × 1)	
		3	2	23 × 19, 12 × 23	4	4 × 9, 4 × 8, 2(7 × 9)	

in some instances with $\xi = 4$, the "best lower bound," the "best feasible solution," and the "gap in %" are reported instead of the unknown optimal value. Remaining columns "MIP iterations," "B&B nodes," and "CPU time" (in seconds) are self-explanatory and state the effort required by the solver to obtain the reported solution. Figure 4 shows the influence of considering leftovers in the reduction of the cost of the objects that need to be purchased to satisfy the overall demand of items along the considered time horizon. As expected, the use of leftovers significantly reduces the cost of the purchased objects. It should be noted that for $\xi = 3$ and $\xi = 4$, the costs of the purchased objects coincide in all the 25 instances. The difference in these two scenarios relies on the value of the remaining leftovers. When $\xi = 3$, there are remaining leftovers in five instances only (namely, instances 1, 2, 3, 5, and 22). When $\xi = 4$, there are remaining leftovers in all instances but instance 4. It can be highlighted the case of instance 14 that has no remaining leftovers when $\xi = 3$ and it has remaining leftovers with value 582 when $\xi = 4$.

As can be seen in Table 3, instances 4, 5, and 9 are infeasible when $\xi = 0$. This is because these instances have instants with ordered items and no available objects, and $\xi = 0$ impair the use of leftovers. The same happens with instance 9 for the case $\xi = 1$ since it has two consecutive periods with ordered items and no available objects. Tables 3 and 4 show that for $\xi \in \{0, 1, 2, 3\}$ the solver was able to detect infeasibility or to find an optimal solution for all the 25 instances, within the CPU time limit. For the case $\xi = 4$ (see Table 5), the CPU time limit was reached for instances 6, 9, 10, 14, 15, 18, 23, and 25. However, it should be noted that the final gap was larger than 1%

Table 2
Number of binary variables (BV), continuous variables (CV), and constraints (CO) of 25 considered instances

	$\xi = 0$			$\xi = 1$			$\xi = 2$			$\xi = 3$			$\xi = 4$		
#Inst.	BV	CV	CO	BV	CV	CO	BV	CV	CO	BV	CV	CO	BV	CV	CO
1	31	34	112	81	82	410	153	162	802	297	354	1498	297	354	1498
2	39	40	143	89	88	423	165	168	823	285	296	1439	509	552	2431
3	63	44	199	115	92	547	211	172	1227	403	300	2947	659	556	4003
4	61	50	258	121	98	588	237	206	1292	461	438	2508	653	662	3404
5	55	52	233	127	108	651	267	256	1507	427	432	2467	587	656	3299
6	146	72	688	276	168	1704	468	296	3448	772	488	6392	1060	744	7512
7	25	32	99	61	80	275	121	160	583	217	288	1071	409	544	1999
8	25	32	99	61	80	275	121	160	583	217	288	1071	409	544	1999
9	108	64	564	204	112	1392	360	208	2568	672	496	4488	1008	880	5976
10	277	84	783	373	132	2171	521	212	4231	713	340	6351	905	596	7279
11	44	44	156	108	100	546	224	212	1202	360	340	1994	584	596	2986
12	29	38	110	85	106	382	161	190	786	281	326	1410	505	614	2434
13	25	38	92	87	114	374	167	214	766	343	422	1566	599	710	2654
14	60	44	184	106	86	482	174	154	898	262	258	1338	374	402	1850
15	52	44	168	108	94	514	212	178	1238	372	314	2294	628	602	3382
16	48	44	204	106	102	494	206	202	1006	374	370	1846	806	802	3574
17	51	44	175	113	104	513	181	172	877	285	276	1421	413	420	1965
18	59	52	251	143	136	735	275	272	1491	483	520	2531	595	664	3043
19	38	40	132	92	90	440	180	174	940	308	310	1580	564	598	2668
20	39	42	142	101	100	466	233	220	1090	425	396	1882	585	540	2490
21	27	42	94	87	114	402	211	274	1034	355	434	1818	643	818	3210
22	46	48	176	112	128	510	224	272	1078	336	400	1678	528	656	2606
23	120	70	515	224	150	1301	436	310	2953	748	598	4873	1084	982	6361
24	55	46	182	109	102	460	217	214	972	377	374	1684	505	502	2212
25	80	60	300	180	140	996	328	252	2100	584	476	3772	920	860	5260

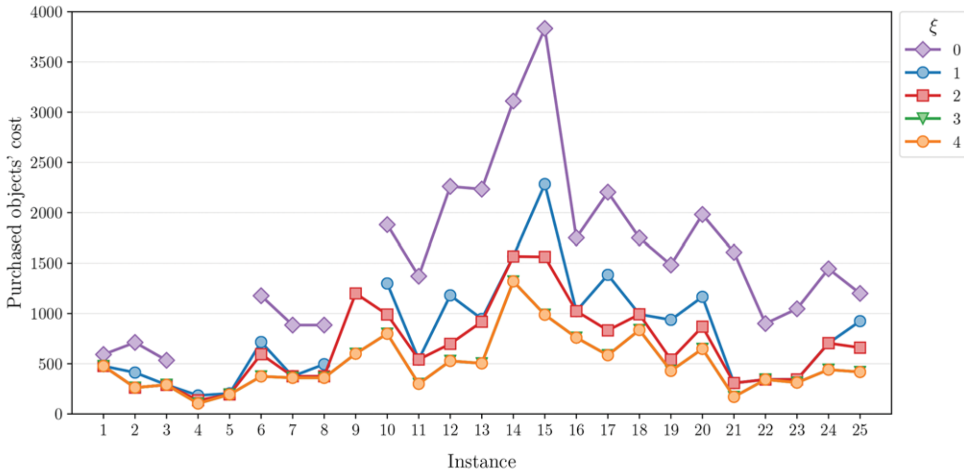


Fig. 4. Graphical representation of the influence of the usage of leftovers in the reduction of the cost of the purchased objects required to satisfy the demand.

Table 3
Description of the solutions found and the effort measurements corresponding to 25 considered instances with $\xi = 0$ and $\xi = 1$

	$\xi = 0$					$\xi = 1$					
	Solutions description		Effort measurements			Solutions description			Effort measurements		
#Inst.	Objective function optimal value	Objects cost	MIP iterations	B&B nodes	CPU time	Objective function optimal value	Objects cost	Leftovers value	MIP iterations	B&B nodes	CPU time
1	775,520	592	7	0	0.20	624,810	477	60	974	122	4.53
2	575,296	712	0	0	0.01	332,758	412	138	73	0	2.02
3	325,206	534	0	0	0.01	176,610	290	0	97	0	0.57
4	Infeasible	–	–	–	–	102,672	184	0	113	0	0.81
5	Infeasible	–	–	–	–	134,232	204	0	80	0	2.31
6	3,789,940	1177	45	0	0.03	2,305,520	716	0	427	0	1.30
7	995,625	885	8	0	0.01	421,875	375	0	77	0	1.42
8	995,625	885	6	0	0.01	556,875	495	0	110	0	1.20
9	Infeasible	–	–	–	–	Infeasible	–	–	–	–	–
10	4,448,565	1881	0	0	0.10	3,072,135	1299	0	62212	7794	24.64
11	3,204,027	1371	5	0	0.01	1,266,654	542	0	108	0	1.15
12	7,314,335	2261	4	0	0.01	3,820,393	1181	142	190	2	2.39
13	11,735,202	2234	4	0	0.01	4,969,338	946	0	100	0	0.62
14	9,678,321	3111	0	0	0.01	4,874,937	1567	0	112	0	1.31
15	15,619,716	3834	0	0	0.01	9,305,016	2284	0	117	0	1.41
16	4,396,761	1751	16	0	0.33	2,571,264	1024	0	185	0	1.76
17	8,207,696	2204	0	0	0.02	5,157,740	1385	0	96	0	0.81
18	4,977,000	1750	32	0	0.40	2,821,248	992	0	133	0	1.65
19	2,835,496	1483	0	0	0.01	1,791,157	937	387	185	0	5.12
20	7,168,894	1982	7	0	0.01	4,217,422	1166	0	143	0	6.31
21	5,367,504	1608	0	0	0.01	1,028,104	308	0	58	0	0.02
22	2,870,100	900	2	0	0.01	1,093,827	343	0	136	0	1.00
23	2,824,806	1047	28	0	0.03	928,112	344	0	303	0	0.73
24	3,761,335	1445	0	0	0.01	1,835,115	705	0	117	0	0.80
25	4,184,400	1200	11	0	0.02	3,221,988	924	0	289	0	1.78

Table 4
Description of the solutions found and the effort measurements corresponding to 25 considered instances with $\xi = 2$ and $\xi = 3$

	$\xi = 2$						$\xi = 3$					
	Solutions description			Effort measurements			Solutions description			Effort measurements		
#Inst.	Objective function optimal value	Objects cost	Leftovers value	MIP iterations	B&B nodes	CPU time	Objective function optimal value	Objects cost	Leftovers value	MIP iterations	B&B nodes	CPU time
1	624,810	477	60	2768	829	5.30	624,810	477	60	10,639	1147	4.11
2	211,696	262	0	177	0	0.44	211,619	262	77	2,376,889	263,942	21.86
3	176,480	290	130	1,483,474	481,658	32.61	176,480	290	130	961,431	257,043	25.87
4	75,888	136	0	469	0	1.03	58,032	104	0	1099	0	1.19
5	127,652	194	0	373	0	2.53	127,597	194	55	2,202,637	479,702	46.13
6	1,915,751	595	149	208,811	33,801	4.69	1,204,280	374	0	48,335	5419	4.97
7	421,656	375	219	753	32	6.58	405,000	360	0	229	0	1.22
8	421,875	375	0	454	0	1.67	405,000	360	0	771	0	1.49
9	4,320,000	1200	0	46,946	2576	2.66	2,160,000	600	0	279,079	6895	7.26
10	2,341,350	990	0	36,240,062	569,320	771.02	1,889,635	799	0	7,009,845	216,734	188.53
11	1,266,477	542	177	17,972	4062	8.69	701,100	300	0	659	0	3.27
12	2,257,888	698	142	281	0	1.53	1,708,080	528	0	407	0	2.37
13	4,822,254	918	0	266	0	1.64	2,647,512	504	0	568	0	0.14
14	4,874,792	1567	145	1783	231	4.82	4,106,520	1320	0	399	0	1.04
15	6,367,662	1563	0	291	0	1.33	4,025,112	988	0	547	0	1.46
16	2,571,018	1024	246	127,651	7755	2.60	1,908,360	760	0	873	0	2.37
17	3,098,368	832	0	317	0	1.65	2,178,540	585	0	404	0	1.64
18	2,821,171	992	77	796	81	9.38	2,377,584	836	0	883	0	2.92
19	1,034,392	541	0	294	0	2.05	820,248	429	0	529	0	0.87
20	3,143,019	869	154	2276	432	3.58	2,343,816	648	0	677	0	0.53
21	1,027,990	308	114	926	73	6.51	567,460	170	0	651	0	4.78
22	1,093,793	343	34	743	37	5.26	1,093,726	343	101	4284	740	5.55
23	928,063	344	49	6,041,800	684,437	35.32	841,776	312	0	112,177	4854	7.45
24	1,834,901	705	214	3391	801	9.20	1,147,923	441	0	483	0	1.08
25	2,301,376	660	44	44,647	9552	6.02	1,457,566	418	0	39,019	2396	3.19

Table 5
Description of the solutions found and the effort measurements corresponding to 25 considered instances with $\xi = 4$

	$\xi = 4$							
	Solutions description					Effort measurements		
#Inst.	Ceiling of best lower bound	Best feasible solution	Objects cost	Leftovers value	Gap (%)	MIP iterations	B&B nodes	CPU time
1	624,810	624,810	477	60	–	10,639	1147	0.84
2	211,619	211,619	262	77	–	2,160,643	201,861	18.03
3	176,480	176,480	290	130	–	841,436	237,602	17.58
4	58,032	58,032	104	0	–	850,790	23,572	15.78
5	127,592	127,592	194	60	–	229,858,101	31,994,580	2987.59
6	1,204,205	1,204,236	374	44	0.00257	159,153,968	14,968,217	>7200.00
7	404,810	404,810	360	190	–	19,283	1844	0.93
8	404,825	404,825	360	175	–	15,865	2084	1.08
9	2,158,783	2,159,724	600	276	0.04357	262,314,681	7,630,137	>7200.00
10	1,143,263	1,889,492	799	143	39.49363	233,353,289	2,425,193	>7200.00
11	700,984	700,984	300	116	–	17,734,163	2,369,734	204.76
12	1,707,878	1,707,878	528	202	–	175,060	27,057	4.11
13	2,647,144	2,647,144	504	368	–	56,098	7948	2.40
14	4,105,120	4,105,938	1320	582	0.01992	246,817,566	16,454,312	>7200.00
15	4,024,518	4,024,714	988	398	0.00487	316,265,168	23,099,236	>7200.00
16	1,907,986	1,907,986	760	374	–	180,021,974	39,120,509	4258.83
17	2,178,286	2,178,286	585	254	–	393,331,330	59,172,386	5388.47
18	2,376,937	2,377,194	836	390	0.01081	226,180,153	28,601,529	>7200.00
19	820,152	820,152	429	96	–	796,992	114,618	11.97
20	2,343,485	2,343,485	648	331	–	14,533,560	2,123,452	170.81
21	567,386	567,386	170	74	–	1479	235	0.87
22	1,093,712	1,093,712	343	115	–	10,229	1594	0.78
23	841,259	841,715	312	61	0.05418	486,159,219	15,462,361	>7200.00
24	1,147,735	1,147,735	441	188	–	30,166,459	3,731,902	316.14
25	1,457,411	1,457,506	418	60	0.00652	169,625,339	14,676,677	>7200.00

in only one instance. The median CPU times for $\xi \in \{0, 1, 2, 3, 4\}$ are 0.01, 1.36, 4.69, 2.92, and 204.76, respectively.

Figures 5–7 show the graphical representation (cutting/packing patterns) of the solutions obtained for (the arbitrarily chosen) instances 6, 7, and 8, varying $\xi \in \{0, 1, 2, 3, 4\}$. Figure 5a shows that, when $\xi = 0$, that is, when leftovers are not considered, four objects that cost 1177 are needed to cut the ordered items, and, of course, there are no leftovers at the end of time horizon. When $\xi = 1$, leftovers can be used only in the period that follows the period in which they were generated. Figure 5b shows that, in this case, only two objects that cost 716 are required. One is bought at instant $p = 0$ and the other at instant $p = 2$. The first one is used to cut the items ordered at instant $p = 0$ and it generates a leftover that is used to cut the items ordered at instant $p = 1$. The same happens at instant $p = 2$, where an object is bought that is used to cut the items ordered at that instant and generates two leftovers that are used to cut the items ordered at instant $p = 3$. Since no object is bough at instant $p = 3$ and $\xi = 1$, there are no leftovers at the end of the time horizon. The case $\xi = 2$ is depicted in Fig. 5c. In this case, the leftovers of the object bought at instant $p = 0$

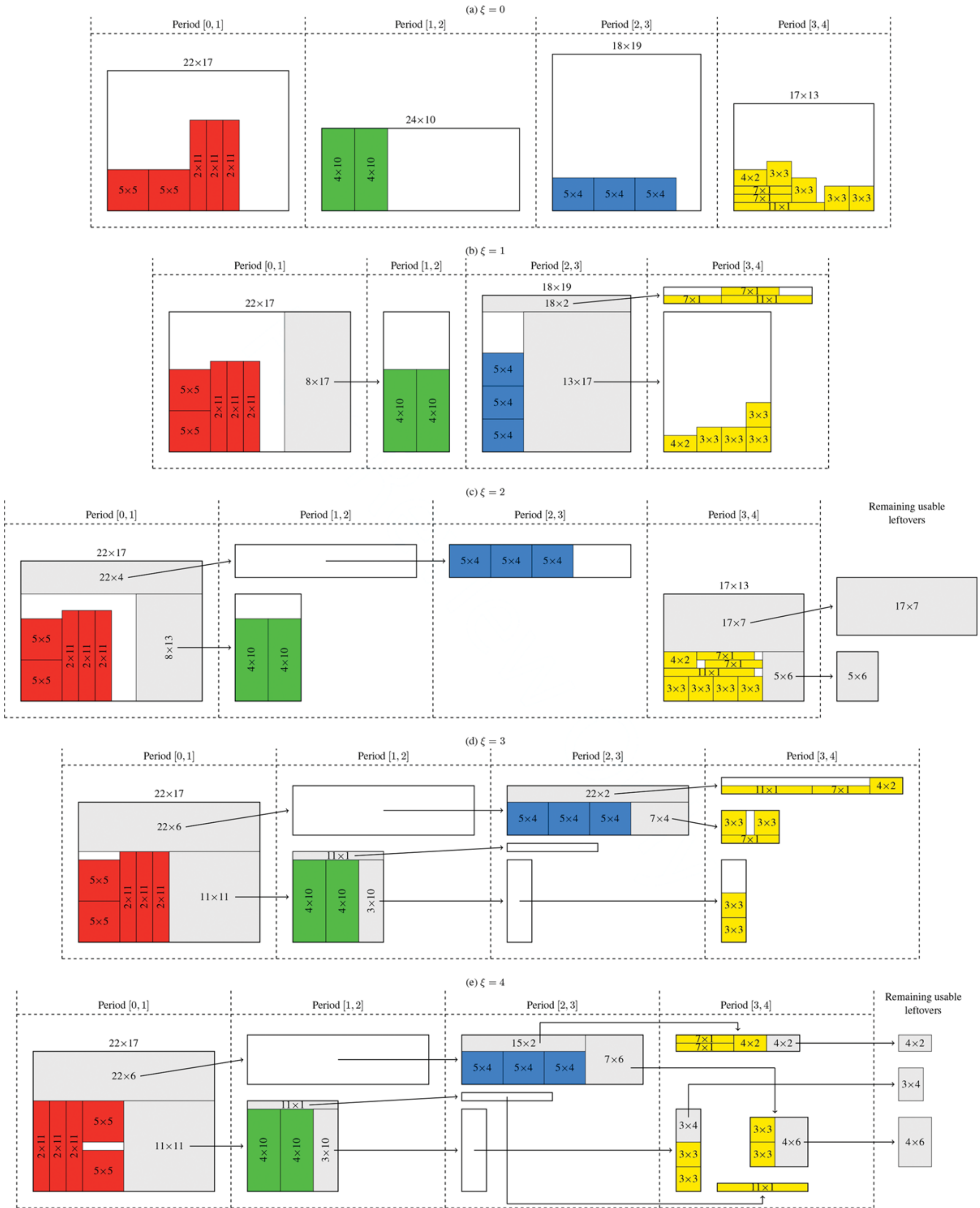


Fig. 5. Graphical representation of the solutions to instance 6 with $\xi \in \{0, 1, \ldots, 4\}$.

are used to cut the items ordered at instants $p = 1$ and $p = 2$. A new object is bought at instant $p = 3$ and two leftovers remain at the end of the time horizon. The two purchased objects cost 595. When $\xi = 3$ (see Fig. 5d), a single object that costs 374 is bought at instant $p = 0$. This object and its leftovers are enough to cut all ordered items (at instants $p = 0, 1, 2, 3$). Since $\xi = 3$ and the only object is bought at instant $p = 0$, there are no leftovers remaining at the end of the time horizon. When $\xi = 4$ (see Fig. 5e), the same object is bought at instant $p = 0$, but the cutting pattern is chosen in order to maximize the value of the leftovers remaining at the end of the time horizon.

Figures 6 and 7 correspond to instances 7 and 8, respectively. These two instances are very similar, the only difference being that they have the cost per unit of area of the two objects with dimensions 10×12 and 12×10 that are available at instant $p = 0$ interchanged. The figures show that, in the case $\xi = 4$, the cutting pattern is such that the value of the remaining leftovers at the end of the horizon is maximized. And this is achieved concentrating the leftovers in the object with a larger cost per unit of area. Note that, in instance 7, the overall area of the remaining leftovers is 95, but its value is 190 (see Table 5), while in instance 8 the overall leftovers' area is 107 with a value of 175 (see Table 5).

As can be seen in Table 1, instance 1 has three periods, while all the other 24 considered instances have four periods. It is very clear from the problem formulation that the number of binary variables in the proposed model depends on the number of periods P and on the number of available objects $\bar{m}_p$ and ordered items n_p at each instant $p = 0, \ldots, P - 1$. Moreover, since, as defined in (1), the number of available objects $\bar{m}_p$ at a given instant p corresponds to the number of purchasable objects m_p plus the number usable leftovers from previous periods, $\bar{m}_p$ depends exponentially on p and on parameter ξ that says for how many periods leftovers of previous periods may be available. This dependency is illustrated in Table 2. Moreover, the results in Tables 3–5 illustrate that, as expected, the solver's effort increases as a function of ξ and that several instances with $\xi = 4$ cannot be solved to optimality within the CPU time limit of two hours. It should also be stressed that, as already mentioned, the goal of the numerical experiments was to analyze the influence of considering leftovers in the overall cost of the purchased objects. Therefore, the set of instances was chosen in such a way that the solver was able to find the optimal solution within an affordable time limit, while it is not a surprise that the solver would not be able to find optimal solutions within a reasonable amount of time for much larger instances.

5. Concluding remarks

Two-dimensional non-guillotine cutting stock problems with leftovers in which leftovers can be generated by two guillotine precuts were considered in this work. In particular, the cutting problem with leftovers introduced in Andrade et al. (2014) was embedded into a multiperiod framework. In this way, objects and leftovers at each period can be better chosen in order to minimize the overall cost of the objects that are required to execute a given set of sorted orders. Some alternative variants of the problem were analyzed. An MIP formulation of the considered problem was introduced and illustrative numerical experiments were presented. On the one hand, since, as expected, practical instances could not be solved to optimality with an exact solver, developing heuristic methods for the introduced problem would be a possible direction for future research. On the other hand, considering uncertainty in the problem data would make the problem closer to practice. A more

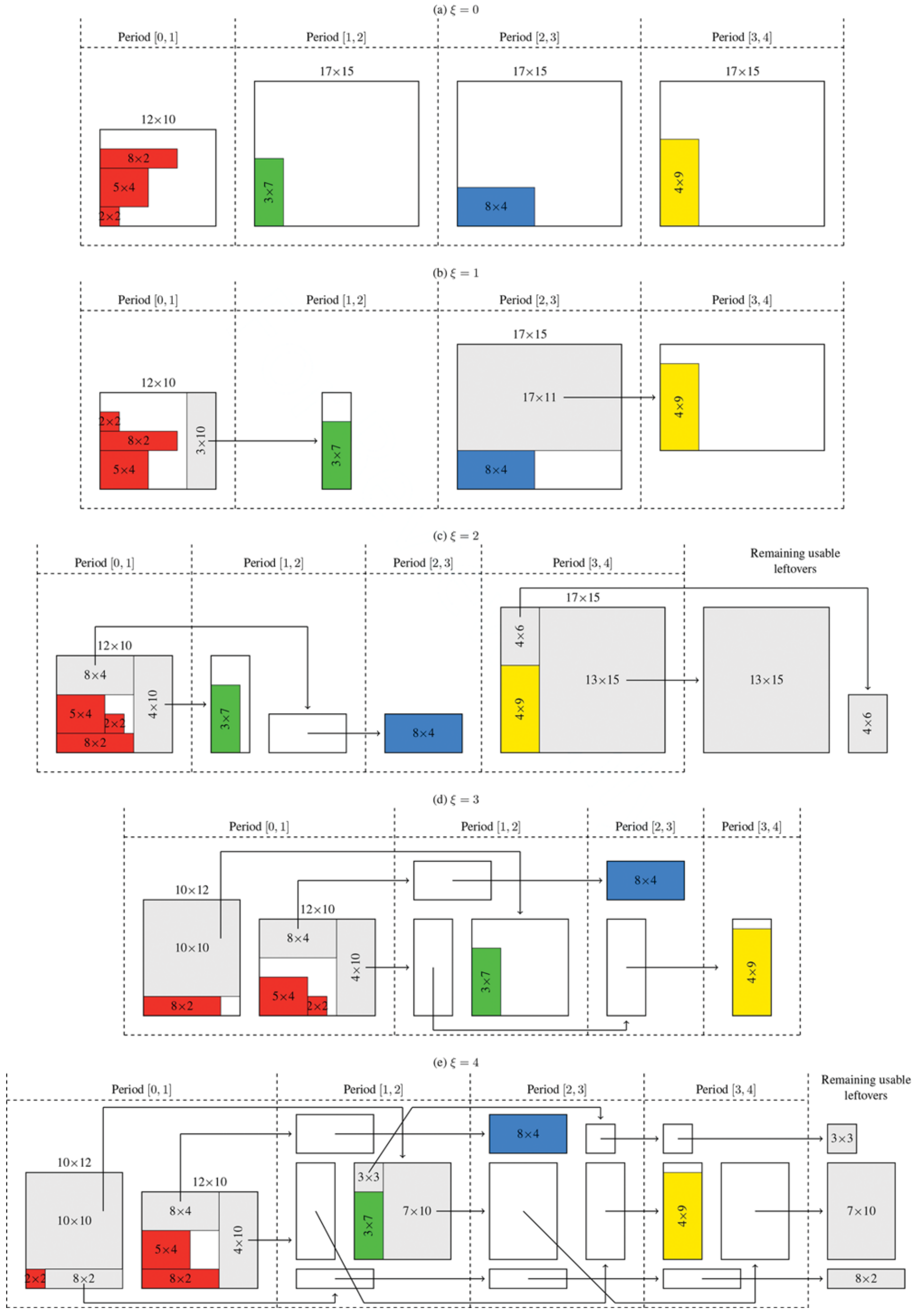


Fig. 6. Graphical representation of the solutions to instance 7 with $\xi \in \{0, 1, \ldots, 4\}$.

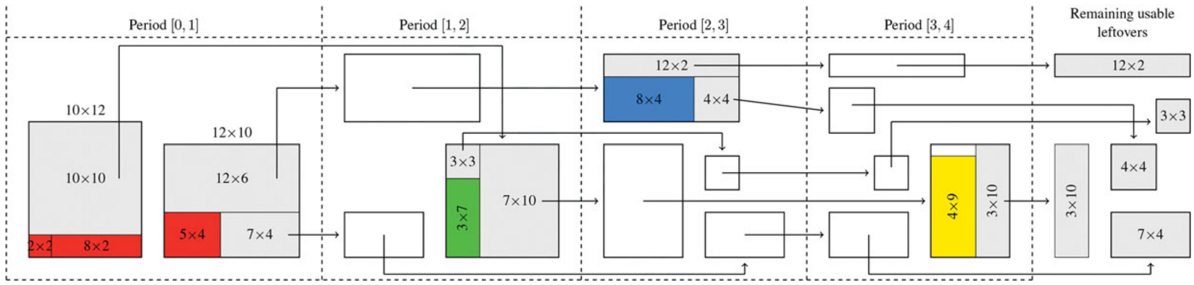


Fig. 7. Graphical representation of the solution to instance 8 with $\xi = 4$.

ambitious goal would be to integrate this multiperiod cutting problem with leftovers with a lot sizing problem in order to simultaneously determine the demanded items that must be placed as orders and their optimal cutting pattern.

Acknowledgments

This work has been partially supported by FAPESP (grants 2013/03447-6, 2013/05475-7, 2013/07375-0, and 2016/01860-1) and CNPq (grants 146110/2013-7, 309517/2014-1, and 306083/2016-7). The authors would like to thank the reviewers for their helpful comments.

References

Alonso, M.T., Alvarez-Valdes, R., Iori, M., Parreño, F., 2019. Mathematical models for multi-container loading problems with practical constraints. *Computers & Industrial Engineering* 127, 722–733.

Andrade, R., Birgin, E.G., 2013. Symmetry-breaking constraints for packing identical orthogonal rectangles within polyhedra. *Optimization Letters* 7, 375–405.

Andrade, R., Birgin, E.G., Morabito, R., 2016. Two-stage two-dimensional guillotine cutting stock problems with usable leftovers. *International Transactions in Operational Research* 23, 121–145.

Andrade, R., Birgin, E.G., Morabito, R., Ronconi, D.P., 2014. MIP models for two-dimensional non-guillotine cutting problems with usable leftovers. *Journal of the Operational Research Society* 65, 1649–1663.

Birgin, E.G., Lobato, R.D., Morabito, R., 2010. An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. *Journal of the Operational Research Society* 61, 306–320.

Birgin, E.G., Lobato, R.D., Morabito, R., 2012. Generating unconstrained two-dimensional non-guillotine cutting patterns by a recursive partitioning algorithm. *Journal of the Operational Research Society* 63, 183–200.

Chen, Q.L., Li, L.P., Cui, Y.D., Chen, Y., Lu, X.Y., 2015. A heuristic for the 3-staged 2D cutting stock problem with usable leftover. In Chan, K., Yeh, J. (eds) *AER-Advances in Engineering Research, Proceedings of the 2015 International Conference on Electrical, Automation and Mechanical Engineering*, Vol. 13, pp. 776–779.

Chen, Y., Song, X., Ouelhadj, D., Cui, Y., 2019. A heuristic for the skiving and cutting stock problem in paper and plastic film industries. *International Transactions in Operational Research* 26, 157–179.

Cherri, A.C., Arenales, M.N., Yanasse, H.H., 2013. The usable leftover one-dimensional cutting stock problem—a priority-in-use heuristic. *International Transactions in Operational Research* 20, 189–199.

Cherri, A.C., Arenales, M.N., Yanasse, H.H., 2014. The one-dimensional cutting stock problem with usable leftovers—a survey. *European Journal of Operational Research* 236, 395–402.

Dempe, S., 2002. *Foundations of Bilevel Programming*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Dyckhoff, H., 1981. A new linear programming approach to the cutting stock problem. *Operations Research* 29, 1092–1104.

Harjunkoski, I., Porn, R., Westerlund, T., Skrifvars, H., 1997. Different strategies for solving bilinear integer non-linear programming problems with convex transformations. *Computers and Chemical Engineering* 21 (Supplement S), S487–S492.

Koch, S., König, S., Wäsher, G., 2009. Integer linear programming for a cutting problem in the wood-processing industry: a case study. *International Transactions in Operational Research* 16, 715–726.

Miettinen, K., 1998. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, MA.

Poldi, K.C., Arenales, M.N., 2010. O problema de corte de estoque unidimensional multiperíodo. *Pesquisa Operacional* 30, 153–174.

Roodman, G.M., 1986. Near-optimal solutions to one-dimensional cutting stock problems. *Computers & Operations Research* 13, 713–719.

Scheithauer, G., 1991. A note on handling residual lengths. *Optimization* 22, 461–466.

Silva, E., Avelos, 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.

Silva, E., Avelos, F., Valério de Carvalho, J.M., 2014. Integrating two-dimensional cutting stock and lot-sizing problems. *Journal of the Operational Research Society* 65, 108–123.

Tomat, L., Gradišar, M., 2017. One-dimensional stock cutting: optimization of usable leftovers in consecutive orders. *Central European Journal of Operations Research* 25, 473–489.

Viegas, J.L., Vieira, S.M., Henriques, E.M.P., Sousa, J.M.C., 2016. Heuristics for three-dimensional steel cutting with usable leftovers considering large time periods. *European Journal of Industrial Engineering* 10, 431–454.

Yanasse, H.H., Morabito, R., 2006. Linear models for 1-group two-dimensional guillotine cutting problems. *International Journal of Production Research* 44, 3471–3491.