Expert Systems With Applications 223 (2023) 119866

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A forward-looking matheuristic approach for the multi-period two-dimensional non-guillotine cutting stock problem with usable leftovers

Ernesto G. Birgin [a,*,1], Oberlan C. Romão [b,1], Débora P. Ronconi [c,1]

[a] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil

[b] Department of Computing and Electronics, University Center of Northern Espírito Santo of the Federal University of Espírito Santo, Rodovia BR 101 Norte, Km 60, Bairro Litorâneo, 29932-540, São Mateus, ES, Brazil

[c] Department of Production Engineering, Polytechnic School, University of São Paulo, Av. Luciano Gualberto, 1380, Cidade Universitária, 05508-010, São Paulo SP, Brazil

ARTICLE INFO

Keywords:
Two-dimensional cutting stock with usable leftovers
Non-guillotine cutting and packing
Multi-period scenario
Forward-looking or looking-ahead approach
Matheuristic

ABSTRACT

In Birgin et al. (2020) the multi-period two-dimensional non-guillotine cutting stock problem with usable leftovers was introduced. At each decision instant, the problem consists in determining a cutting pattern for a set of ordered items using a set of objects that can be purchased or can be leftovers of previous periods; the goal being the minimization of the overall cost of the objects up to the considered time horizon. Among solutions with minimum cost, a solution that maximizes the value of the leftovers at the end of the considered horizon is sought. A forward-looking matheuristic approach that applies to this problem is introduced in the present work. At each decision instant, the objects and the cutting pattern that will be used is determined, taking into account the impact of this decision in future states of the system. More specifically, for each potentially used object, an attempt is made to estimate the utilization rate of its leftovers and thereby determine whether the object should be used or not. The introduced approach is compared with an exact off-the-shelf commercial solver and a myopic technique. Numerical experiments show the efficacy of the proposed approach.

1. Introduction

In this paper, we consider the multi-period two-dimensional non-guillotine cutting stock problem with usable leftovers. In the problem, P periods of time denoted by $[s-1,s]$ for $s=1,\dots,P$ are considered; period $[s-1,s]$ corresponding to $t_{s-1} \le t \le t_s$, where $t_0 < t_1 < \cdots < t_P$ are given decision time instants. Small rectangular pieces of varying sizes (named items) can be ordered at any instant t between t_0 and t_{P-1}. However, assuming the discrete time convention, if an item is ordered at an instant t such that $t_{s-1} < t \le t_s$ for some $s \in \{1,\dots,P-1\}$, then it is assumed the item was ordered at instant t_s. All items ordered at instant t_s must be produced between t_s and t_{s+1} and delivered at instant t_{s+1}. Raw material is available in the form of large rectangular purchasable pieces (named purchasable objects) or as usable leftovers of previous periods, i.e. parts of objects purchased at previous periods that were not used to produce items. (Remains of the cutting process can be classified as usable leftovers or can be discarded as scrap. Usable leftovers will be formally defined in Section 2, but roughly speaking they cannot be very old and must satisfy size constraints.) At each instant t_s, ordered items are known and the problem consists in selecting objects to be purchased and existent leftovers to produce all ordered items. The cutting pattern of each object (leftover or purchased) must also be determined. The problem is said to be two-dimensional because it involves the width and the height of items and objects; while it is said to be non-guillotine because cuts are not restricted to be guillotine cuts. Assuming the material is anisotropic, rotations of the items are not allowed. Objects as well as leftovers can produce new leftovers. The amount of leftovers in stock is maintained under control with a parameter $\xi \in \{0,1,\dots,P\}$ that determines that parts (leftovers, leftovers of leftovers, etc.) of an object purchased at instant t_s can only be used at instants $t_{s+1},\dots,t_{s+\xi}$. (If $\xi=0$, the problem has no leftovers at all; while, if $\xi=1$, leftovers can only be used in the period immediately following the period in which they were generated.) The goal is to minimize the overall cost of objects purchased to produce all orders from instant t_0 to instant t_{P-1} and, among the minimum cost solutions, to choose one in which the value of the usable leftovers remaining at instant t_P (end of the considered time horizon) is maximized.

* Corresponding author.
E-mail addresses: egbirgin@ime.usp.br (E.G. Birgin), oberlan.romao@ufes.br (O.C. Romão), dronconi@usp.br (D.P. Ronconi).

[1] All the authors of this paper contributed equally in all phases of the development of the work.

https://doi.org/10.1016/j.eswa.2023.119866
Received 6 February 2022; Received in revised form 21 February 2023; Accepted 11 March 2023
Available online 22 March 2023
0957-4174/

As introduced in Birgin et al. (2020) and described in the paragraph above, the problem includes two hierarchically ordered objectives related to the cost of raw material and keeps the stock under control by constraining it, with parameter ξ, to be proportional to incoming orders. Alternatively, the cost of inventories could have been included in the objective function. In this paper, we chose to follow the formulation proposed in Birgin et al. (2020), which assumes that there is installed and available capacity for storage and handling of stocks and minimizes the raw material cost by taking into account the leftovers.

In the current work, we propose a forward-looking matheuristic to solve medium- and large-sized instances of the described problem. Forward-looking strategies have already been used with great success in different types of discrete optimization problems; see, for example, Ronconi and Powell (2010), Powell (2007) and the references therein. In a training phase, the method attempts to estimate the proportion of each generated usable leftover that will be effectively used to produce items ordered in forthcoming periods. With this information, at a given period, a more expensive object can be purchased if the estimated future use of its leftovers points to future savings. A subproblem is solved per period. The decision variables determine the objects the must be purchased, the leftovers from previous periods that will be used, and their cutting pattern. All ordered items must be produced; and the goal is to minimize an objective function that, by discounting the cost of leftovers that are assumed to be used in the near future to produce ordered items within the considered time horizon, minimize the effective cost of the raw material needed to produce the items ordered in the period. The estimation of the effective usage of leftovers being generated, that is required to estimate the actual cost of the raw material, constitutes the forward-looking ingredient of the method. At the end of each training cycle, the estimated utilization proportion of each leftover is compared with its actual utilization proportion, and the estimate is updated. The updating rule and the stopping criterion ensure that the number of training cycles is finite.

The proposed method is calibrated with the instances with four periods considered in Birgin et al. (2020); and then evaluated on a new set of instances with four, eight, and twelve periods. The performance of the method is compared with a myopic approach on the new set of thirty instances with up to twelve periods. For the new (small) instances with four periods, an additional comparison with CPLEX is also presented. The myopic approach differs with the forward-looking approach only in the objective function being minimized at each period. While the forward-looking approach considers the possible future use of letfovers, the myopic approach greedily minimizes the cost of the objects necessary to produce the ordered items of the period. The problem includes a parameter that tells for how many periods, after being generated, a leftover is available for use, thus keeping the stock under control. The larger the durability of the leftovers, the greater the opportunity to save with the acquisition of purchasable objects. Experiments show that the forward-looking approach outperforms the myopic approach by a large extent and that, the greater the number of periods or the larger the durability of usable leftovers, the greater the advantage.

The problem considered in the present work was proposed in Birgin et al. (2020), where a mixed integer linear programming model was introduced and instances with up to four periods were solved using CPLEX. However, no solution method has yet been proposed to deal with larger instances of the problem. The single-period version of the problem was considered in Andrade et al. (2014), where a discussion related to alternative definitions of usable leftovers was presented. Several papers in the literature, many of them based on real-world applications, address the one-dimensional cutting stock problem with usable leftovers; see the pioneers' works (Roodman, 1986; Scheithauer, 1991) and the more recent works (Ali et al., 2021; Baykasoglu & Özbel, 2021; Cherri et al., 2013, 2014; do Nascimento et al., 2022; Poldi & Arenales, 2010; Tomat & Gradišar, 2017). On the other hand, only a few publications tackle the two-dimensional case considered in the present work.

In all publications dedicated to the one-dimensional problem mentioned in the previous paragraph, a multi-period scenario is considered and a single threshold determines whether a cutting pattern leftover is disposed of as trim-loss or is a usable leftover. In particular, Tomat and Gradišar (2017) focuses on determining the optimal amount of usable leftovers that should be kept in stock in order to make good use of the raw material and at the same time minimize the cost of stock handling. In Cherri et al. (2013), a heuristic that prioritizes the use of leftovers in order to control their stock quantity is presented. A rolling horizon scheme for the same problem is proposed in Poldi and Arenales (2010). The subproblem of each period is solved with a simplex method with column generation and different strategies are considered in order to obtain integer solutions through rounding. A survey that reviews published studies up to 2014 can be found in Cherri et al. (2014). A recent work (do Nascimento et al., 2022) integrates the problem with the lot-sizing problem. In the problem under consideration, it is possible to bring forward the production of items with known demand in a future period. A relax-and-fix approach is proposed that solves the subproblems with a simplex method with column generation. Other recent works present practical applications in the marble industry (Baykasoglu & Özbel, 2021) and in the use of leftover piping in construction (Ali et al., 2021).

Exact and non-exact two- and three-stage two-dimensional cutting stock problems with leftovers are considered in Silva et al. (2010). In the considered problem, a single item is cut from a raw material object at a time, through one or two guillotine cuts, generating zero, one, or two "residual objects". A MILP model that extends the one-cut model presented in Dyckhoff (1981) for the one-dimensional cutting stock problem is introduced; and numerical experiments solving real-world instances of the furniture industry and instances from the literature are presented. MILP models are solved with CPLEX. On the one hand, the goal is minimizing the number of cuts. On the other hand, several extensions, such as minimizing the number of used raw material objects (that are all of the same type), minimizing the length of the cuts, minimizing waste, allowing rotations, and considering multiple type of objects are also considered. One of the extensions, that points to attributing a value to the leftovers, opens the possibility of embedding the considered problem in a multi-period framework, as its was later done by the same authors in Silva et al. (2014). In Silva et al. (2014), the problem is integrated with the lot-sizing problem with the aim of minimizing a total cost that includes material, waste and storage costs. In the problem under consideration, anticipating the production of items maximizes raw material utilization while incurring stock costs; and a balance between these conflicting objectives is sought by minimizing their pricing. Two MILP models that do not depend on cutting patterns generation and two heuristics based on the industrial practice are presented. In contrast to the problem considered in the present work, at each period, two-stage non-exact cutting patterns are generated. In a brief contribution (Chen et al., 2015), a single-period problem with three-stage cutting patterns is considered in which the leftovers consist of remnants of the first cutting stage, the objective being to minimize the difference between the object cost and the value of the usable leftovers generated. A real-world multi-period three-dimensional cutting problem related to the supply of steel blocks in the metalworking is considered in Viegas et al. (2016). Since remnants from one period can be used to produce items ordered in future periods, the problem considers leftovers; the objective being to keep stock growth under control. For the problem at hand, constructive heuristic procedures are proposed.

The rest of this paper is organized as follows. Section 2 provides a formal description of the multi-period two-dimensional non-guillotine cutting stock problem with leftovers. Section 3 introduces the proposed matheuristic with a looking-ahead feature. Section 4 presents numerical experiments. Conclusions and lines for future research are given in the last section.

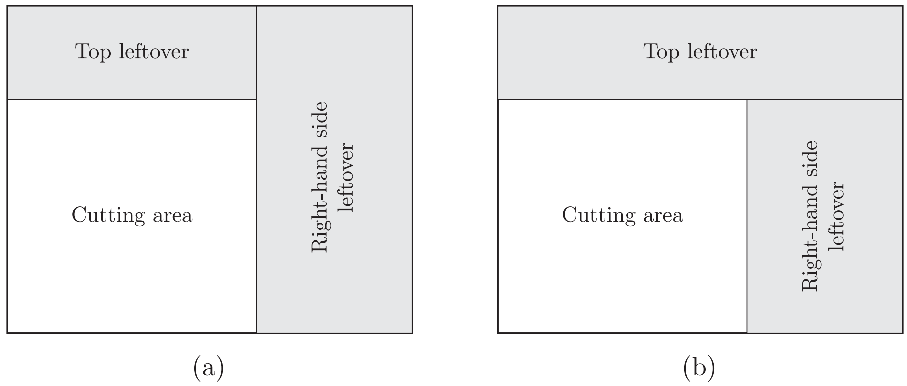

Fig. 1. Pictures (a) and (b) illustrate the two possible ways in which two leftovers can be generated from an object by performing a vertical and a horizontal guillotine pre-cut. In case (a), the vertical guillotine pre-cut is made first; while, in case (b), the horizontal guillotine pre-cut is made first.

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

In this section, the multi-period two-dimensional non-guillotine cutting stock problem with usable leftovers is described; and its mixed integer linear programming formulation introduced in Birgin et al. (2020) is presented. The (single-period) two-dimensional non-guillotine cutting stock problem with leftovers was introduced in Andrade et al. (2014) and extended to the multi-period framework in Birgin et al. (2020). One of the main features of the problem is that, when an object is used to cut items from it, two leftovers are obtained by performing a couple of guillotine pre-cuts on the object that separate the leftovers from the cutting area of the object (region from where the items will be cut); see Fig. 1. Of course, the leftovers are optional and either one or both can be empty, i.e. have zero area. It should be noted that, regardless of whether or not there are leftovers and, if there are, how they are separated from the object, the items are cut from the cutting area of the object in a non-guillotine pattern. Given a catalogue of items, we say a leftover is usable if it can fit at least an 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. 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. (2016) and Andrade et al. (2014) for other definitions of leftovers in two-dimensional problems. Andrade et al. (2014) includes a detailed description of the single-period version of the problem, with several examples. Unlike the multi-period model presented in Birgin et al. (2020), the model introduced in this section considers time instants s from p to P. The possibility of choosing the initial and final instants of the model gives the necessary flexibility to formulate subproblems in algorithms of the rolling horizon type as the one that will be presented later.

Let p and P satisfying $p < P$ be the first and the last instant to be considered, respectively. For each instant $s = p, \ldots, P-1$, there are given m_s purchasable objects $\mathcal{O}_{sj}$ with width W_{sj}, height H_{sj}, and cost c_{sj} per unit of area ($j = 1, \ldots, m_s$) and a set of n_s ordered items $\mathcal{I}_{si}$ with width w_{si} and height h_{si} ($i = 1, \ldots, n_s$). A catalogue composed by d items $\bar{\mathcal{I}}_i$ with width $\bar{w}_i$ and height $\bar{h}_i$ ($i = 1, \ldots, d$) is also given. A parameter $\xi \in [0, P-p]$ says that leftovers generated within a period $[s, s+1)$ remain valid up to period $[s+\xi, s+\xi+1)$. By definition, each object generates two leftovers. This means that the number of objects at instant s is given by

$$\bar{m}_s = m_s + 2\,\hat{m}_{s-1} \text{ for } s = p, \ldots, P, \tag{1}$$

where

$$\hat{m}_s = \sum_{\ell=0}^{\min\{s-p,\xi-1\}} 2^{\ell} m_{s-\ell}, \text{ for } s = p, \ldots, P-1, \tag{2}$$

stands for the number of objects that, at period $[s, s+1)$, generate leftovers, $\hat{m}_{p-1} = 0$ (i.e. no leftovers coming from previous periods at the first considered instant $s = p$), and $m_P = 0$ (i.e. no purchasable objects at the last considered instant $s = P$). 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. The problem consists in minimizing the overall cost of the purchasable objects required to produce the items ordered at instants $p, \ldots, P-1$ making use of leftovers; and, among all solutions with minimum cost, maximizing the value of the usable leftovers at instant P. See Figs. 2 and 3. Fig. 2 describes a toy instance of the problem; while Fig. 3 exhibits two different feasible solutions.

Purchasable objects $\mathcal{O}_{sj}$ ($s = p, \ldots, P-1$, $j = 1, \ldots, m_s$) have a given cost c_{sj} per unit of area. The value of an usable leftover is given by its area times its cost per unit of area; and the cost per unit of area of a leftover corresponds to the cost per unit of area of the purchasable object from which the leftover comes from. In order to make this relation, we associate with each (purchasable or leftover) object $\mathcal{O}_{sj}$ ($s = p, \ldots, P$, $j = 1, \ldots, \bar{m}_s$) an expiration date e_{sj} in such a way that, if $\mathcal{O}_{sj}$ is a purchasable object, we define $e_{sj} = \xi$; while if $\mathcal{O}_{sj}$ is a leftover then we define e_{sj} as the expiration date of the object from which it comes from reduced by one. Clearly, $e_{sj} \geq 0$, since objects with null expiration date do not generate leftovers. Let $j_1^s \leq j_2^s \leq \cdots \leq j_{\hat{m}_s}^s$ be the indices of the $\hat{m}_s$ objects that generate leftovers in the period $[s, s+1)$; and let us define that, at instant $s+1$, objects $\mathcal{O}_{s+1,m_{s+1}+2k-1}$ and $\mathcal{O}_{s+1,m_{s+1}+2k}$ correspond to the "top leftover" and to the "right-hand-side leftover" of object $\mathcal{O}_{s,j_k^s}$, respectively. Thus, $c_{s+1,m_{s+1}+2k-1} = c_{s+1,m_{s+1}+2k} = c_{s,j_k^s}$ and $e_{s+1,m_{s+1}+2k-1} = e_{s+1,m_{s+1}+2k} = e_{s,j_k^s} - 1$. The relevant costs are the costs c_{Pj} ($j = m_P + 1, \ldots, \bar{m}_P$) that correspond to the value (per unit of area) of the leftovers available at instant P, i.e. at the end of the considered time horizon, that are the leftovers whose value must be maximized. For a given instant s ($s = p, \ldots, P-1$) and the expiration dates e_{sj} of the $\bar{m}_s$ objects available at the instant, the $\hat{m}_s \leq \bar{m}_s$ indices $j_1^s, j_2^s, \ldots$ of the objects that potentially generate leftovers can be computed as follows. Start with $k = 0$ and, for j from 1 to $\bar{m}_s$, if $e_{sj} > 0$ then increase k by one and set $j_k^s = j$. Finish by setting $\hat{m}_s = k$.

The description of the problem's variables follows. Variables $v_{sij} \in \{0, 1\}$ ($s = p, \ldots, P-1$, $j = 1, \ldots, \bar{m}_s$, $i = 1, \ldots, n_s$) assign items to objects ($v_{sij} = 1$ if item $\mathcal{I}_{si}$ is assigned to object $\mathcal{O}_{sj}$; and $v_{sij} = 0$ otherwise). Variables $u_{sj} \in \{0, 1\}$ ($s = p, \ldots, P-1$, $j = 1, \ldots, m_s$) identify whether at least an item is assigned to object $\mathcal{O}_{sj}$ or not ($u_{sj} = 1$ and $u_{sj} = 0$,

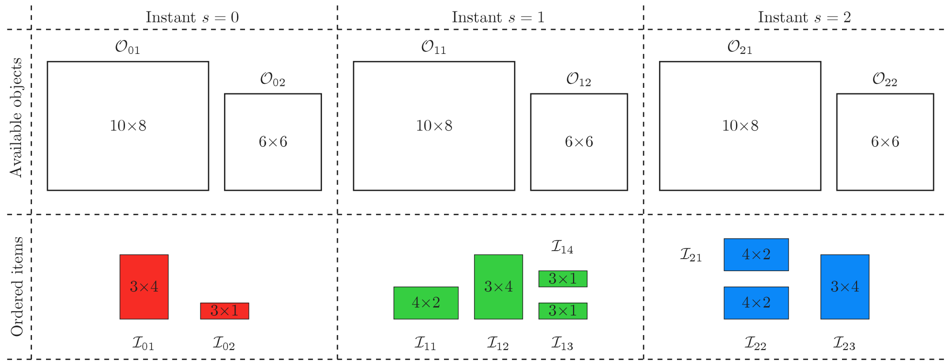

Fig. 2. Illustration of a small instance with $p = 0$, $P = 3$, and $\xi = P - p = 3$, meaning that usable leftovers generated at any period remain usable up to instant P. The picture shows the available purchasable objects and the ordered items at each instant $s \in \{0, 1, 2\}$. The numbers of available purchasable objects and ordered items at each instant are given by $m_0 = m_1 = m_2 = 2$ and $n_0 = 2$, $n_1 = 4$ and $n_2 = 3$, respectively. The cost per unit of area of all the objects is one (i.e. $c_{01} = c_{02} = c_{11} = c_{12} = c_{21} = c_{22} = 1$) and the catalogue with $d = 1$ item is composed by an item with $\bar{w}_1 = 3$ and $\bar{h}_1 = 1$.

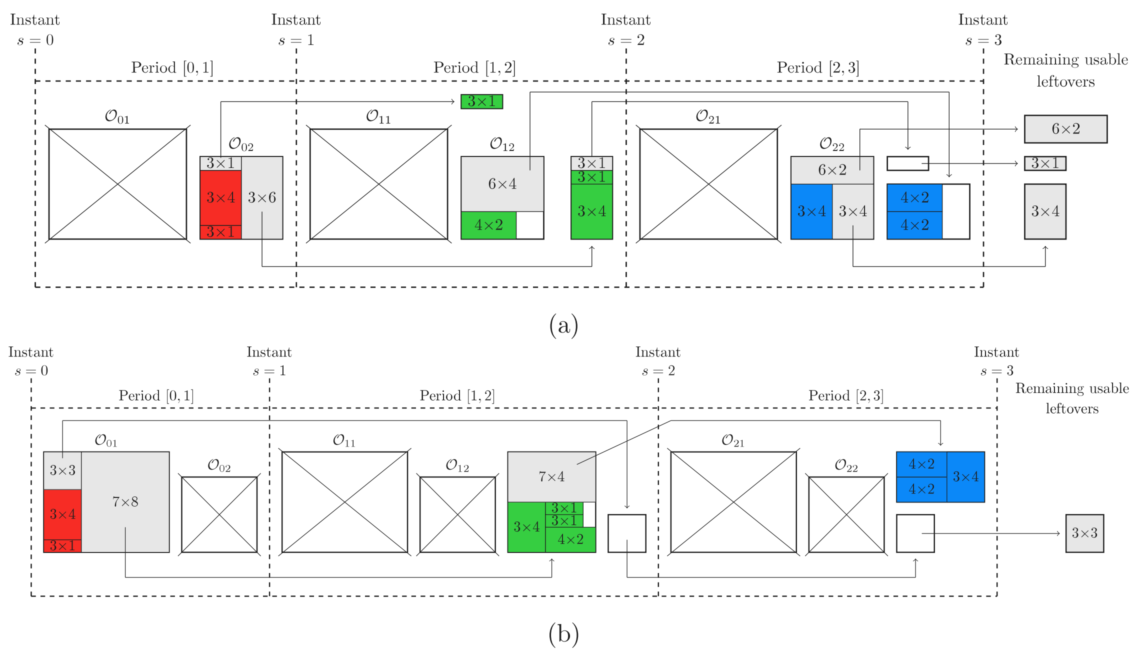

Fig. 3. Illustration of two solutions that, at each period, may cut ordered items from purchasable objects or from usable leftovers from previous periods. (a) Greedy solution obtained by a myopic method that, *at each decision instant*, minimizes the cost of the purchasable objects required to cut the ordered items of that instant, assuming that usable leftovers from previous periods are free. (b) Solution with minimum total cost of the required purchasable objects and, in addition, maximum value of the usable leftovers at instant $P = 3$. The cost of the purchased objects in the solution in (a) is 108; while the same cost is 80 in (b).

respectively). Variables $\eta_{sj} \in \{0, 1\}$ $(s = p, \ldots, P - 1,\ j = 1, \ldots, \bar{m}_s)$ determine if the vertical pre-cut that separates the cutting area from the leftover in object $\mathcal{O}_{sj}$ is made before the horizontal pre-cut ($\eta_{sj} = 1$) or if the horizontal pre-cut precedes the vertical pre-cut ($\eta_{sj} = 0$). Variables t_{sj} and $r_{sj} \in \mathbb{R}$ $(s = p, \ldots, P-1,\ j = 1, \ldots, \bar{m}_s)$ determine the height of the top leftover and the width of the right-hand-side leftover of object $\mathcal{O}_{sj}$, respectively. Variables $\bar{W}_{sj}$ and $\bar{H}_{sj} \in \mathbb{R}$ $(s = p, \ldots, P,\ j = 1, \ldots, \bar{m}_s)$ represent the width and the height of object $\mathcal{O}_{sj}$. (This is relevant to the objects that are leftovers of objects purchased at previous periods, since the dimensions of purchasable objects are constant, i.e. $\bar{W}_{sj} = W_{sj}$ and $\bar{H}_{sj} = H_{sj}$ for every s whenever $1 \le j \le m_s$.) Variables $\pi_{sii'}$ and $\tau_{sii'} \in \{0, 1\}$ $(s = p, \ldots, P - 1,\ i = 1, \ldots, n_s,\ i' = i + 1, \ldots, n_s)$ are auxiliary variables used to avoid the overlapping between items. Variables $\gamma_j \in \mathbb{R}$ $(j = 1, \ldots, \bar{m}_P)$ are related to the value of the area of the leftovers at instant P, i.e. at the end of the considered time horizon. Variables $\theta_{j\ell} \in \{0, 1\}$ and $\omega_{j\ell} \in \mathbb{R}$ $(j = 1, \ldots, \bar{m}_P,\ \ell = 1, \ldots, L)$ are auxiliary variables used to linearize the computation of these areas (product of the leftovers variable dimensions), where $L = \lfloor \log_2(\hat{W}) \rfloor + 1$, $\hat{W} = \max\{W_{sj} \mid s = p, \ldots, P-1, j = 1, \ldots, m_s\}$, and, for further reference, $\hat{H} = \max\{H_{sj} \mid s = p, \ldots, P - 1, j = 1, \ldots, m_s\}$. The auxiliary variables $\zeta_{ji} \in \{0, 1\}$ $(j = 1, \ldots, \bar{m}_P,\ i = 1, \ldots, d)$ are used to nullify the value of the area of a leftover at instant P if it cannot fit any item from the catalogue.

The problem consists in minimizing

$$\left(\sum_{s=p}^{P-1} \sum_{j=1}^{m_s} c_{sj} W_{sj} H_{sj} \right) \left(\sum_{s=p}^{P-1} \sum_{j=1}^{m_s} c_{sj} W_{sj} H_{sj} u_{sj} \right) - \sum_{j=m_P+1}^{\bar{m}_P} c_{Pj} \gamma_j \tag{3}$$

subject to

$$\sum_{j=1}^{\bar{m}_s} v_{sij} = 1,\ s = p, \ldots, P - 1,\ i = 1, \ldots, n_s, \tag{4}$$

$$u_{sj} \ge v_{sij},\ s = p, \ldots, P - 1,\ j = 1, \ldots, \bar{m}_s,\ i = 1, \ldots, n_s, \tag{5}$$

$$u_{sj} \le \sum_{i=1}^{n_s} v_{sij},\ s = p, \ldots, P - 1,\ j = 1, \ldots, \bar{m}_s, \tag{6}$$

$$0 \le t_{sj} \le \bar{H}_{sj} \text{ and } 0 \le r_{sj} \le \bar{W}_{sj},\ j = 1, \ldots, \bar{m}_s, \tag{7}$$

$$\frac{1}{2}w_{si} \le x_{si} \le \bar{W}_{sj} - r_{sj} + (1 - v_{sij})\hat{W} - \frac{1}{2}w_{si},$$
$$s = p, \dots, P-1, i = 1, \dots, n_s, j = 1, \dots, \bar{m}_s, \quad (8)$$

$$\frac{1}{2}h_{si} \le y_{si} \le \bar{H}_{sj} - t_{sj} + (1 - v_{sij})\hat{H} - \frac{1}{2}h_{si},$$
$$s = p, \dots, P-1, i = 1, \dots, n_s, j = 1, \dots, \bar{m}_s, \quad (9)$$

$$\begin{array}{rcccl}
0 & \le & \bar{H}_{s+1,\ell_1} & \le & \hat{H}u_{sj}, \\
t_{sj} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_1} & \le & t_{sj} + (1-u_{sj})\hat{H}, \\
0 & \le & \bar{W}_{s+1,\ell_1} & \le & \hat{W}u_{sj}, \\
\bar{W}_{sj} - r_{sj} - (1-\eta_{sj})\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} - r_{sj} + (1-\eta_{sj})\hat{W} + (1-u_{sj})\hat{W}, \\
\bar{W}_{sj} - \eta_{sj}\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} + \eta_{sj}\hat{W} + (1-u_{sj})\hat{W}, \\
0 & \le & \bar{W}_{s+1,\ell_2} & \le & \hat{W}u_{sj}, \\
r_{sj} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_2} & \le & r_{sj} + (1-u_{sj})\hat{W}, \\
0 & \le & \bar{H}_{s+1,\ell_2} & \le & \hat{H}u_{sj}, \\
\bar{H}_{sj} - (1-\eta_{sj})\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} + (1-\eta_{sj})\hat{H} + (1-u_{sj})\hat{H}, \\
\bar{H}_{sj} - t_{sj} - \eta_{sj}\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} - t_{sj} + \eta_{sj}\hat{H} + (1-u_{sj})\hat{H},
\end{array} \quad (10)$$

for $s = p, \dots, P-1$ and $j = j_k^s \le m_s$ for $k = 1, \dots \hat{m}_s$, with $\ell_1 = m_{s+1} + 2k - 1$ and $\ell_2 = m_{s+1} + 2k$,

$$\begin{array}{rcccl}
\bar{H}_{sj} - \hat{H}u_{sj} & \le & \bar{H}_{s+1,\ell_1} & \le & \bar{H}_{sj} + \hat{H}u_{sj}, \\
t_{sj} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_1} & \le & t_{sj} + (1-u_{sj})\hat{H}, \\
\bar{W}_{sj} - \hat{W}u_{sj} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} + \hat{W}u_{sj}, \\
\bar{W}_{sj} - r_{sj} - (1-\eta_{sj})\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} - r_{sj} + (1-\eta_{sj})\hat{W} + (1-u_{sj})\hat{W}, \\
\bar{W}_{sj} - \eta_{sj}\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} + \eta_{sj}\hat{W} + (1-u_{sj})\hat{W}, \\
0 & \le & \bar{W}_{s+1,\ell_2} & \le & \hat{W}u_{sj}, \\
r_{sj} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_2} & \le & r_{sj} + (1-u_{sj})\hat{W}, \\
0 & \le & \bar{H}_{s+1,\ell_2} & \le & \hat{H}u_{sj}, \\
\bar{H}_{sj} - (1-\eta_{sj})\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} + (1-\eta_{sj})\hat{H} + (1-u_{sj})\hat{H}, \\
\bar{H}_{sj} - t_{sj} - \eta_{sj}\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} - t_{sj} + \eta_{sj}\hat{H} + (1-u_{sj})\hat{H},
\end{array} \quad (11)$$

for $s = p, \dots, P-1$ and $j = j_k^s > m_s$ for $k = 1, \dots \hat{m}_s$, with $\ell_1 = m_{s+1} + 2k - 1$ and $\ell_2 = m_{s+1} + 2k$,

$$\begin{array}{rcl}
x_{si} - x_{si'} & \ge & \frac{1}{2}(w_{si} + w_{si'}) - \hat{W}\left[(1-v_{sij}) + (1-v_{si'j}) + \pi_{sii'} + \tau_{sii'}\right], \\
-x_{si} + x_{si'} & \ge & \frac{1}{2}(w_{si} + w_{si'}) - \hat{W}\left[(1-v_{sij}) + (1-v_{si'j}) + \pi_{sii'} + (1-\tau_{sii'})\right], \\
y_{si} - y_{si'} & \ge & \frac{1}{2}(h_{si} + h_{si'}) - \hat{H}\left[(1-v_{sij}) + (1-v_{si'j}) + (1-\pi_{sii'}) + \tau_{sii'}\right], \\
-y_{si} + y_{si'} & \ge & \frac{1}{2}(h_{si} + h_{si'}) - \hat{H}\left[(1-v_{sij}) + (1-v_{si'j}) + (1-\pi_{sii'}) + (1-\tau_{sii'})\right],
\end{array} \quad (12)$$

for $s = p, \dots, P-1$, $j = 1, \dots, \bar{m}_s$, $i = 1, \dots, n_s$, $i' = i+1, \dots, n_s$,

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

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

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

and

$$\bar{W}_{Pj} = \sum_{\ell=1}^{L} 2^{\ell-1}\theta_{j\ell} \text{ for } j = m_P + 1, \dots, \bar{m}_P. \quad (16)$$

The objective function (3) is given by the cost of the used *purchasable* objects multiplied by a strict upper bound on the value of the leftovers at instant P minus the value of the leftovers at that instant. Assuming integrality of the constants that define the instance (see Birgin et al., 2020, §3.7), this composition has the desired effect of minimizing the cost of the purchased objects and, among solutions with the same cost, maximizing the value of the leftovers at instant P. Constraints (4) say that each item must be assigned to exactly one object. Constraints (5) and (6) say that an object $\mathcal{O}_{sj}$ is used (i.e. $u_{sj} = 1$) if and only if at least an item is allocated to the object. At a first glance, since the cost of the used objects is being minimized, constrains (6) may appear to be superfluous. However, forcing $u_{sj} = 0$ when no item is assigned to object $\mathcal{O}_{sj}$ prevents purchasing and cutting an object to which no item is being assigned in period s. Constraints (7) define the height t_{sj} of the top leftover and the width r_{sj} of the right-hand-side leftover of object $\mathcal{O}_{sj}$. Constraints ((8),(9)) assume, without loss of generality, that objects have its bottom-left corner in the origin of the Cartesian two-dimensional space. Constraints ((8),(9)) say that if an item $\mathcal{I}_{si}$ is assigned to an object $\mathcal{O}_{sj}$, that has dimensions $\bar{W}_{sj}$ and $\bar{H}_{sj}$, then the center (x_{si}, y_{si}) of the item must be placed within the cutting area of the object that goes from $(0,0)$ to $(\bar{W}_{sj} - r_{sj}, \bar{H}_{sj} - t_{sj})$. Moreover, the constraints say the center of each item must be far from the borders of the cutting area, so the whole item can be placed within the object's cutting area. In constraints (10), restrictions on the dimensions of the leftovers of purchasable objects with positive expiration date are given; while in (11) the same is done with the dimensions of leftovers of objects that are leftovers of previous periods. The difference is that, in the first case, leftovers of a purchasable object must have null dimensions if the purchasable object is not used (purchased); while, in the second case, if an object that is a leftover is not used and its expiration date is strictly positive, then it must pass to the next instant as its own top or right-hand-side leftover. Constraints (12) model the non-overlapping of items assigned to the same object. Constraints ((13)–(16)) model the value γ_j of the jth leftover of the last instant P, i.e. object $\mathcal{O}_{Pj}$. Recall that, in case a leftover can fit at least an item from the catalogue, its value is given by its area (product of its variable dimensions) times the value per unit of area of the purchasable object that generated the leftover. Otherwise, the value of the leftover is null. (See Birgin et al., 2020, §3.7.1 for details.) In ((13)–(16)), the index j starts from $m_P + 1$. This is the same as saying that it starts at 1, since $m_P = 0$ by definition. However, we opted by writing this way because it simplifies the re-definition of the meaning of variables γ in the next section. Note also that variables ω, θ, ζ, and γ, differently from all other variables in the model, do not have an index s that relates them to an instant of the multi-period scenario. This is because they all refer to the last instant P. Note that the *areas* of the leftovers of the last instant of the considered horizon play a fundamental role in the objective function (3); while for all other instants (including instant P) only the (variable) dimensions of the leftovers are required, but not their areas.

3. Forward-looking proposed heuristic

The mixed integer linear programming (MILP) problem (3)–(16) will be named $\mathcal{M}(p,P)$ from now on. This notation allow us to refer to the single-period problem $\mathcal{M}(\kappa,\kappa+1)$ for some $\kappa \in \{p,\dots,P-1\}$. In problem $\mathcal{M}(\kappa,\kappa+1)$, it is assumed that (a) all decisions of instants $s=p,\dots,\kappa-1$ have already been taken; (b) quantities and dimensions of the ordered items and available objects (that may be purchasable or leftovers from previous periods) of instant κ are known; and (c) the last instant of the considered horizon is pushed back and artificially considered as if it were $P=\kappa+1$. Thus, the single-period problem $\mathcal{M}(\kappa,\kappa+1)$ coincides with the single-period problem introduced in Andrade et al. (2014). This means that problem $\mathcal{M}(\kappa,\kappa+1)$ consists in determining a cutting pattern to produce all items ordered at instant κ minimizing the cost of the purchased objects and, among solutions with minimum cost, choosing one that maximizes the value of the leftovers at instant $\kappa+1$. The particularity of $\mathcal{M}(\kappa,\kappa+1)$ with respect to the single-period problem introduced in Andrade et al. (2014) is that in $\mathcal{M}(\kappa,\kappa+1)$ there are some objects that can be used for free. This is because the summation in (3) goes from 1 up to m_κ; meaning that the costs of objects numbered from $m_\kappa+1$ up to $\bar{m}_\kappa$, that are the leftovers of previous periods, are not included in the objective function. Special attention must also be given to the role of variables γ_j in $\mathcal{M}(\kappa,\kappa+1)$. On the one hand, in $\mathcal{M}(p,P)$, their indices goes from 1 (because $m_P=0$ by definition) to $\bar{m}_P$ and they represent the areas of the leftovers at instant P. On the other hand, in $\mathcal{M}(\kappa,\kappa+1)$, since P is redefined as if it were $\kappa+1$, the indices of variables γ go from $m_{\kappa+1}+1$ to $\bar{m}_{\kappa+1}$; and variables γ represent the areas of the leftovers at instant $\kappa+1$.

If we assume that the available computational capacity is enough to solve (with an exact commercial solver) instances with no more than a single period, a heuristic approach to tackle the original multi-period problem must be considered. At each instant κ, a decision has to be made. The decision consists in selecting a set of objects (between the m_κ purchasable objects $\mathcal{O}_{\kappa j}$ for $j=1,\dots,m_\kappa$ or leftovers $\mathcal{O}_{\kappa j}$ for $j=m_\kappa+1,\dots,\bar{m}_\kappa$ from previous periods) and a cutting pattern to produce, along period $[\kappa,\kappa+1)$, the n_κ items ordered at instant κ. The simplest (matheuristic) approach would be to solve the single-period problem $\mathcal{M}(\kappa,\kappa+1)$, for $\kappa=p,\dots,P-1$. Substituting P by $\kappa+1$ in (3), we have that the objective function of problem $\mathcal{M}(\kappa,\kappa+1)$ is given by

$$\left(\sum_{s=p}^{\kappa}\sum_{j=1}^{m_s} c_{sj}W_{sj}H_{sj}\right)\left(\sum_{s=p}^{\kappa}\sum_{j=1}^{m_s} c_{sj}W_{sj}H_{sj}u_{sj}\right)-\sum_{j=m_{\kappa+1}+1}^{\bar{m}_{\kappa+1}} c_{\kappa+1,j}\gamma_j. \tag{17}$$

Since in problem $\mathcal{M}(\kappa,\kappa+1)$ it is assumed that all decisions of instants $s=p,\dots,\kappa-1$ have already been taken, we have that u_{sj} for $s=p,\dots,\kappa-1$ and $j=1,\dots,m_\kappa$ are constant. Thus, minimizing (17) is equivalent to minimizing

$$C_\kappa \sum_{j=1}^{m_\kappa} c_{\kappa j}W_{\kappa j}H_{\kappa j}u_{\kappa j}-\sum_{j=m_{\kappa+1}+1}^{\bar{m}_{\kappa+1}} c_{\kappa+1,j}\gamma_j, \tag{18}$$

where, as in (3),

$$C_\kappa=\sum_{s=p}^{\kappa}\sum_{j=1}^{m_s} c_{sj}W_{sj}H_{sj}$$

is a constant. Note that C_κ corresponds to the total cost of all purchasable objects existent from the first instant p up to instant κ. Therefore, it is a strict upper bound on the value of the leftovers that could have been generated up to instant $\kappa+1$. Thus, multiplying the first summation in (18) by C_κ has the desired effect of making one unit of this summation to be more relevant that the whole second summation in (18). It is in this way that the cost of the used purchasable objects is minimized and, among solutions with minimum cost, a solution that maximizes the value of the leftovers at the end of the considered horizon, in this case instant $\kappa+1$, is sought. Note that this interpretation requires the first summation in (18) to assume integer values only; see Andrade et al. (2014) for details.

The main drawback of a myopic/greedy strategy like the one described above is that the overall cost is not being minimized at all. This strategy was used to find the solution depicted in Fig. 3(a) to the instance described in Fig. 2. Its flaw is to ignore the effect in the future of the decisions made at each instant κ. Fig. 3(b) shows that, by buying a more expensive object at instant $\kappa=0$, a better solution can be found. In addition, note that, at each instant κ, the number of available objects m_κ is finite. If we redefine $m_1=0$ for the instance in Fig. 2 (i.e. no purchasable objects available at instant $\kappa=1$), then the choice of purchasing the small object $\mathcal{O}_{02}$ at instant $\kappa=0$ produces an infeasible solution. This is because the 3×6 leftover of $\mathcal{O}_{02}$ is not enough to produce the items ordered at $\kappa=1$ and, since we redefined $m_1=0$, no other object is available at $\kappa=1$. So, the myopic approach is unable to find a feasible solution to the modified instance.

Assume that we are at an instant κ and that at that instant there are two different objects (one cheaper and smaller and another more expensive but larger) that can be used to produce the n_κ ordered items. Buying the cheapest object would be the myopic choice. However, assume that buying and using the more expensive object produces two leftovers that, by being used in forthcoming periods, produce an overall saving. Quantifying this saving and using it to decide which object to buy at instant κ is the looking-ahead strategy we are looking for. An optimistic view would consist in subtracting from the cost of each object the value of its leftovers. We say this view is optimistic because it assumes that 100% of the object's leftovers will be used to produce items (and, thus, savings) in forthcoming periods. In a more realistic view, each leftover has a different utilization rate that depends on its dimensions and on the ordered items in the forthcoming periods.

At any instant $\kappa+1$, objects $\mathcal{O}_{\kappa+1,j}$ with index j between $m_{\kappa+1}+1$ and $m_{\kappa+1}+2m_\kappa$ correspond to the $2m_\kappa$ leftovers of the m_κ purchasable objects that were available at instant κ. Therefore, at instant κ, γ_{2j-1} and γ_{2j} correspond to the area of the two leftovers of the purchasable object $\mathcal{O}_{\kappa j}$ for $j=1,\dots,m_\kappa$ (nullified when the object is not purchased or when the leftover does not fit any item from the catalogue). Thus, if object $\mathcal{O}_{\kappa j}$ is used, then its optimistic amortized cost, that assumes that 100% of its leftovers will be used, is given by

$$c_{\kappa j}W_{\kappa j}H_{\kappa j}u_{\kappa j}-c_{\kappa j}\gamma_{2j-1}-c_{\kappa j}\gamma_{2j}. \tag{19}$$

The value of (19) is null if object $\mathcal{O}_{\kappa j}$ is not used because in this case $u_{\kappa j}=\gamma_{2j-1}=\gamma_{2j}=0$. If utilization rates $\delta_{\kappa,2j-1},\delta_{\kappa,2j}\in[0,1]$ for $j=1,\dots,m_\kappa$ were known, then we would be able to compute, at instant κ, the more realistic amortized cost

$$c_{\kappa j}W_{\kappa j}H_{\kappa j}u_{\kappa j}-c_{\kappa j}\left(\delta_{\kappa,2j-1}\gamma_{2j-1}+\delta_{\kappa,2j}\gamma_{2j}\right) \tag{20}$$

of using object $\mathcal{O}_{\kappa j}$ to produce the ordered items. Since we need the summation of costs to assume integer values, we would approximate (20) by

$$c_{\kappa j}W_{\kappa j}H_{\kappa j}u_{\kappa j}-\lfloor c_{\kappa j}\left(\delta_{\kappa,2j-1}\gamma_{2j-1}+\delta_{\kappa,2j}\gamma_{2j}\right)\rfloor. \tag{21}$$

However, since γ_{2j-1} and γ_{2j} ($j=1,\dots,m_\kappa$) are variables of the problem, (21) cannot be included in the objective function. (It is not a linear function of continuous and integer variables.) Thus, we need new integer variables λ_j ($j=1,\dots,m_\kappa$) and constraints

$$\lambda_j \le c_{\kappa j}\left(\delta_{\kappa,2j-1}\gamma_{2j-1}+\delta_{\kappa,2j}\gamma_{2j}\right)\ \text{ for } j=1,\dots,m_\kappa; \tag{22}$$

so we can write the approximation (21) of (20) as

$$c_{\kappa j}W_{\kappa j}H_{\kappa j}u_{\kappa j}-\lambda_j. \tag{23}$$

We call (23) the amortized cost of object $\mathcal{O}_{\kappa j}$. Thus, including estimations of the leftovers utilization rates, the objective function (18) of problem $\mathcal{M}(\kappa,\kappa+1)$ can be substituted by

$$C_\kappa \sum_{j=1}^{m_\kappa}\left(c_{\kappa j}W_{\kappa j}H_{\kappa j}u_{\kappa j}-\lambda_j\right)-\sum_{j=m_\kappa+1}^{\bar{m}_{\kappa+1}} c_{\kappa+1,j}\gamma_j. \tag{24}$$

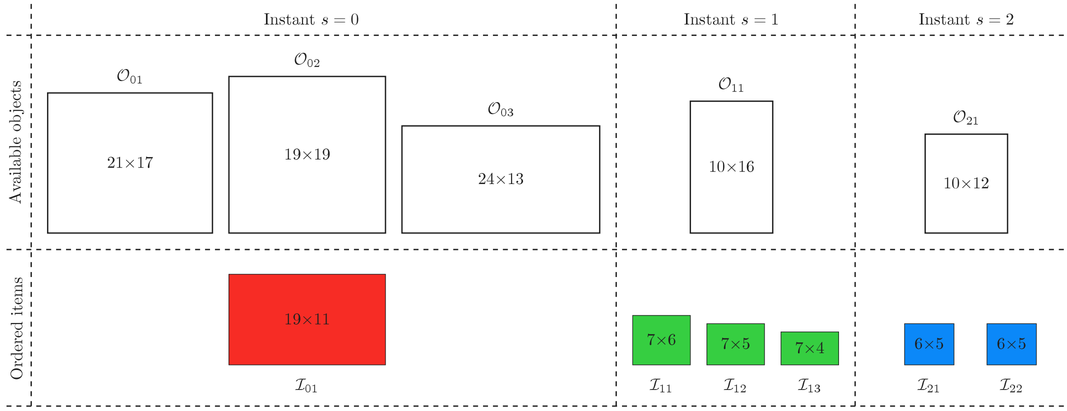

Fig. 4. Illustration of a small instance with $p = 0$, $P = 3$. The figure displays the available purchasable objects and the ordered items at each instant $s \in \{p, \ldots, P-1\}$.

We call $\mathcal{M}(\delta; \kappa, \kappa+1)$, the single-period problem $\mathcal{M}(\kappa, \kappa+1)$ in which the objective function is replaced with (24) and constraints (22) are included. Note that (22) and, in consequence (24), depends on the unknown constants $\delta_{\kappa,2j-1}$ and $\delta_{\kappa,2j}$ for $j = 1, \ldots, m_\kappa$.

Let us illustrate the idea of amortized costs with an example. Fig. 4 displays the available purchasable objects and the ordered items of a small instance with $p = 0$, $P = 3$, and $\xi = P - p = 3$, meaning that usable leftovers generated at any period remain usable up to instant P. The picture shows the available purchasable objects and the ordered items at each instant $s \in \{0, 1, 2\}$. The numbers of available purchasable objects and ordered items at each instant are given by $m_0 = 3$, $m_1 = m_2 = 1$ and $n_0 = 1$, $n_1 = 3$ and $n_2 = 2$, respectively. The cost per unit of area of all the objects is one (i.e. $c_{01} = c_{02} = c_{03} = c_{11} = c_{21} = 1$) and the catalogue with $d = 2$ item is composed by two items with $\bar{w}_1 = 7$, $\bar{h}_1 = 4$, $\bar{w}_2 = 6$, and $\bar{h}_2 = 5$.

At instant $s = 0$, item $\mathcal{I}_{01}$ can be assigned to any of the three available purchasable objects $\mathcal{O}_{01}$, $\mathcal{O}_{02}$, or $\mathcal{O}_{03}$. Dashed regions in Fig. 5(a–c) represent the usable leftovers in each possible assignment. In case (b) there is only a top usable leftover simply because $W_{02} = w_{01}$. In case (a) there is also a top usable leftover only. This is because the right-hand-side leftover has width $W_{02} - w_{01} < \min\{\bar{w}_1, \bar{w}_2\}$. Thus, it cannot fit any item of the catalogue and, therefore, it is *not* usable. In case (c), the situation described in case (a) occurs for both, the top and the right-hand-side leftovers; thus none of them are usable. Since all the three objects have a unitary cost per unit of area (i.e. $c_{01} = c_{02} = c_{03} = 1$), purchasing objects $\mathcal{O}_{01}$, $\mathcal{O}_{02}$, and $\mathcal{O}_{03}$ costs $W_{01} \times H_{01} = 21 \times 17 = 357$, $W_{02} \times H_{02} = 19 \times 19 = 361$, and $W_{03} \times H_{03} = 24 \times 13 = 312$, respectively. The greedy choice mandates to buy object $\mathcal{O}_{03}$, that is the cheapest one. However, assuming that usable leftovers will be 100% used to produce items in forthcoming periods and reducing the value of the leftovers from the cost of their respective objects, we obtain, for the configurations depicted in Fig. 5, the amortized costs $357 - 21 \times 6 = 231$ and $361 - 19 \times 8 = 209$ for objects $\mathcal{O}_{01}$ and $\mathcal{O}_{02}$, respectively. The amortized cost of object $\mathcal{O}_{03}$ whose usage generates no usable leftovers coincides with its actual cost. Thus, the optimistic forward-looking approach would recommend to purchase object $\mathcal{O}_{02}$.

If the myopic approach is applied to the instance of Fig. 4, then the solution found is to purchase object $\mathcal{O}_{03}$ at instant $s = 0$ and objects $\mathcal{O}_{11}$ and $\mathcal{O}_{21}$ at instants $s = 1$ and $s = 2$, respectively. This solution has an overall cost of 592 and has no usable leftovers at instant $s = 3$. If the optimistic forward-looking approach, that assumes that 100% of the usable leftovers will be used in forthcoming periods, is used, then the solution found is the one illustrated in Fig. 6(a). (To simplify the presentation, unused objects are not being displayed in the figure.) In this solution, the object with the smallest amortized cost is chosen at instant $s = 0$, i.e. object $\mathcal{O}_{02}$. At instant $s = 1$, object $\mathcal{O}_{11}$ is purchased and ordered items are produced from the purchased object and from the leftover of the previous period. At instant $s = 2$ no object is purchased and the ordered items are produced from a leftover of the leftover of the object bought at instant $s = 0$. The overall cost of the solution is 521 and a leftover with value 70 remains available at instant $P = 3$. (This solution is clearly better than the solution obtained by the myopic approach.) However, it can be noted that the assumption that 100% of the leftover of object $\mathcal{O}_{02}$ would be used in the next periods turned out to be false. In fact, the leftover of area 152 was used to produce items whose areas totalize 102, i.e. an utilization rate of $102/152 \approx 0.67$. If we consider this utilization rate for object $\mathcal{O}_{02}$, then its amortized cost for the configuration depicted in Fig. 5(b) becomes $361 - 102 = 259$. The amortized cost of object $\mathcal{O}_{01}$ (for the configuration in Fig. 5(a)) remains the same, i.e. 231, since there is no new information to update the presumed utilization rate of 100% of its usable leftover. The amortized cost of object $\mathcal{O}_{03}$ (for the configuration in Fig. 5(a)) continues being 312 as well. Thus, if the problem is solved once again, object $\mathcal{O}_{01}$ is chosen at instant $s = 0$ to produce the ordered items of instant $s = 0$. Then, its leftover is used to produce all ordered items of instant $s = 1$; and object $\mathcal{O}_{21}$ is purchased to produce the items ordered at instant $s = 2$. This solution, depicted at Fig. 6(b), has an overall cost of 477 and it has no usable leftovers at instant $s = 3$. In this solution, the actual utilization rate of the leftover of object $\mathcal{O}_{02}$ is $314/357 \approx 0.88$; which increases its amortized cost for the configuration depicted in Fig. 5(b) from 231 to $357 - \lfloor (314/357) \times 126 \rfloor = 247$. Anyway, it continues to be the cheapest purchasable object at instant $s = 0$. Thus, a new cycle would produce the same solution.

The proposed forward-looking matheuristic approach consists in a sequence of training cycles. In each cycle, the $P - p$ single-period problems $\mathcal{M}(\delta, \kappa, \kappa+1)$ for $\kappa = p, \ldots, P-1$ are solved with fixed values of $\delta_{\kappa,2j-1}$ and $\delta_{\kappa,2j}$ for $\kappa = p, \ldots, P-1$ and $j = 1, \ldots, m_\kappa$. In the 0th cycle, $\delta^0_{\kappa,2j-1} = \delta^0_{\kappa,2j} = \delta_{\text{ini}}$ for all κ and j, where $\delta_{\text{ini}} \in [0,1]$ is a given constant. At the end of the ηth cycle, it is possible to compute the actual fractions $f^\eta_{\kappa,2j-1}$ and $f^\eta_{\kappa,2j}$ of each of the two leftover $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2j-1}$ and $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2j}$ of a purchasable object $\mathcal{O}_{\kappa j}$ that were effectively used to produce items in forthcoming periods for all κ and j. Note that here we are talking about items directly produced from the leftovers $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2j-1}$ and $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2j}$ and also about items produced from leftovers of these leftovers up to ξ periods after purchasing the purchasable object $\mathcal{O}_{\kappa j}$. Thus, each $\delta^\eta_{\kappa,2j-1}$ and $\delta^\eta_{\kappa,2j}$ can be updated using $f^\eta_{\kappa,2j-1}$ and $f^\eta_{\kappa,2j}$. In particular, we define

$$\delta^{\eta+1}_{\kappa,2j-1} = (1-\sigma^\eta)\delta^\eta_{\kappa,2j-1} + \sigma^\eta f^\eta_{\kappa,2j-1} \quad \text{and} \quad \delta^{\eta+1}_{\kappa,2j} = (1-\sigma^\eta)\delta^\eta_{\kappa,2j} + \sigma^\eta f^\eta_{\kappa,2j}, \tag{25}$$

where $\sigma \in (0,1)$ is a given constant and σ^η means σ to the power of η. This means that, at the end of the ηth cycle, new estimations $\delta^{\eta+1}_{\kappa,2j-1}$ and $\delta^{\eta+1}_{\kappa,2j}$ of the utilization rates of the two leftovers of object $\mathcal{O}_{\kappa j}$ for all κ and j are computed as convex combination (parameterized by σ^η) of their previous values $\delta^\eta_{\kappa,2j-1}$ and $\delta^\eta_{\kappa,2j}$ and their actual values $f^\eta_{\kappa,2j-1}$ and $f^\eta_{\kappa,2j}$ in the solution found in the current cycle. Since consecutive

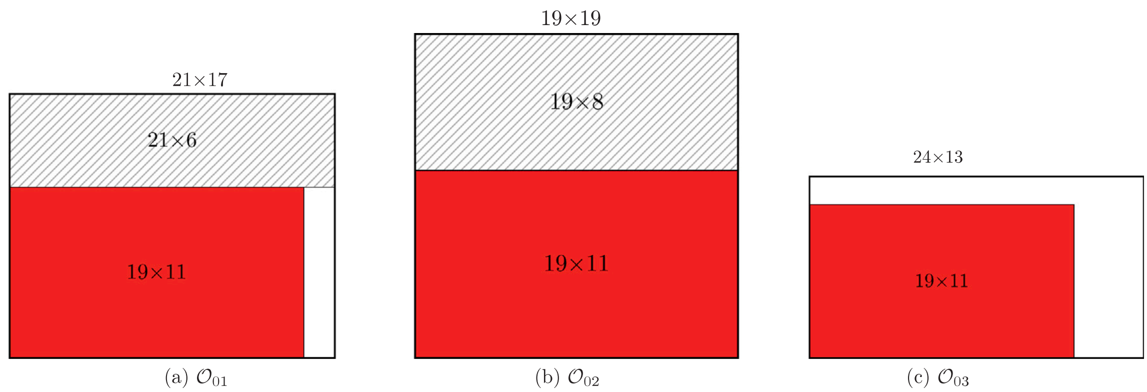

Fig. 5. Dashed regions represent the usable leftovers in the assignment of item $\mathcal{I}_{01}$ to the three purchasable objects available at instant $s = 0$.

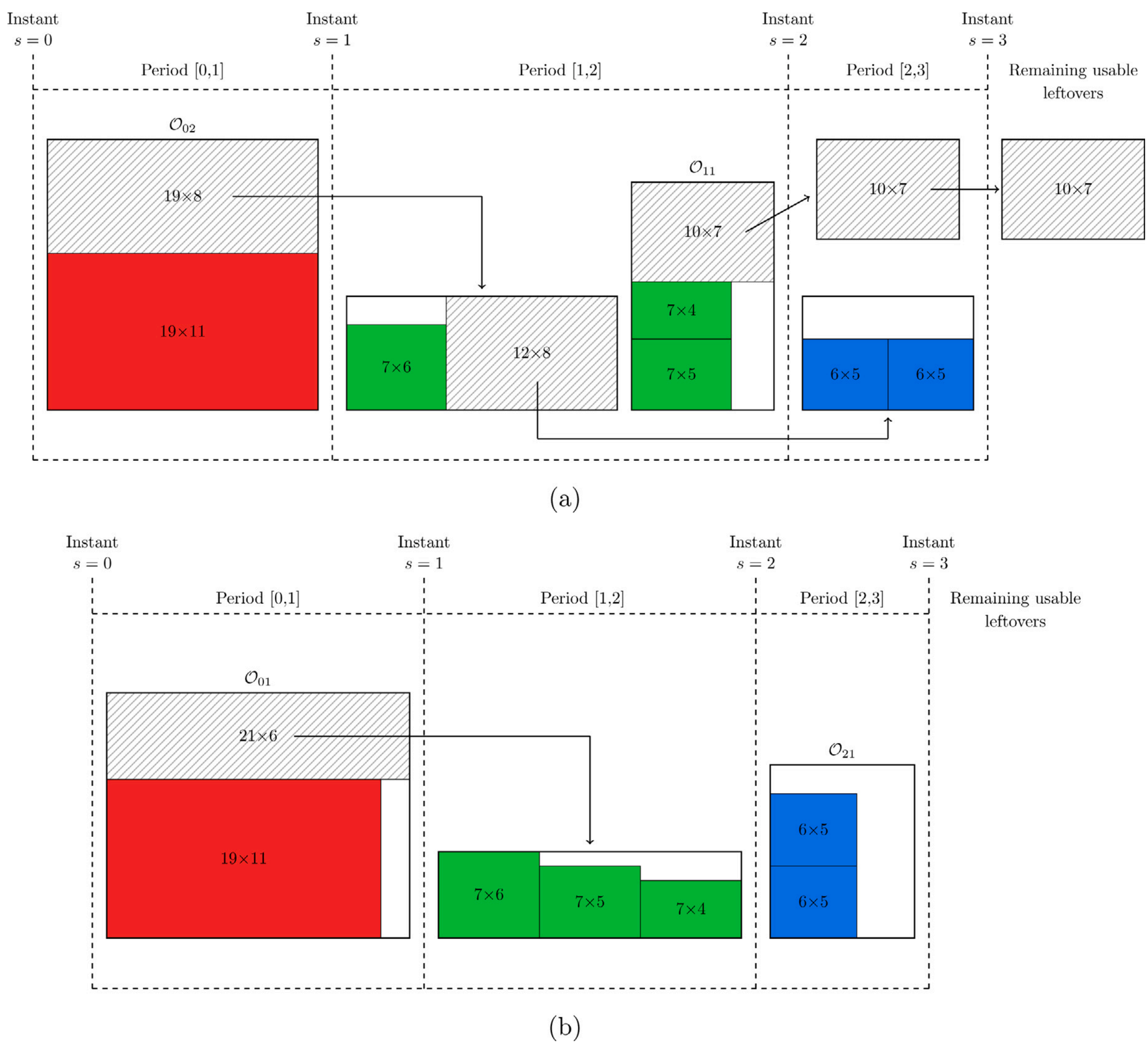

Fig. 6. Different feasible solutions to the instance of Fig. 4. (a) Solution obtained with the optimistic forward-looking approach in which it is assumed that 100% of each usable leftover is used to produce items in forthcoming periods. (b) Solution obtained with an adaptive forward-looking approach that cycles updating the utilization rate of the leftovers.

cycles with the same values of δ's produce the same solution, it makes sense to use

$$\max_{\{\kappa=p,\ldots,P-1,j=1,\ldots,m_\kappa\}} \left\{ \left|\delta^{\eta+1}_{\kappa,2j-1} - \delta^{\eta}_{\kappa,2j-1}\right|, \left|\delta^{\eta+1}_{\kappa,2j} - \delta^{\eta}_{\kappa,2j}\right| \right\} \le \epsilon, \quad (26)$$

where $\epsilon > 0$ is a given constant, as a stopping criterion.

The forward-looking approach considers the utilization rates of the top and the right-hand-side leftovers of purchasable objects. We say these are first-order leftovers. In opposition, when a leftover is a leftover of a leftover, we say it is a high-order leftover. When an item is produced from a first-order leftover, its area plays a role in the utilization rate of the first-order leftover itself. On the other hand, when an item is produced from a high-order leftover, its area plays a role in the utilization rate of the first-order leftover that is the ancestor of the used high-order leftover. Therefore, computing the utilization rate of the first-order leftovers requires to keep track of their successor leftovers or, equivalently, to keep track of the ancestors of the high-order leftovers. Assume we are in the ηth cycle of the forward-looking approach and that the current instant is instant κ. After having solved the single-period problem $\mathcal{M}(\delta^\eta, \kappa, \kappa+1)$ we proceed as follows. (The supra-index η will be omitted for simplicity.) Let $j_1 \le j_2 \le \cdots \le j_{\hat{m}_\kappa}$ be the indices of the $\hat{m}_\kappa$ objects that can generate leftovers, that correspond to the indices j of objects $\mathcal{O}_{\kappa j}$ $(j = 1,\ldots,\bar{m}_\kappa)$ such that $e_{\kappa j} > 0$. Note that if $j_k \le m_\kappa$, then $\mathcal{O}_{\kappa,j_k}$ is a purchasable object and its leftovers are first-order leftovers, while if $m_\kappa < j_k \le \bar{m}_\kappa$, then $\mathcal{O}_{\kappa,j_k}$ is a leftover and its leftovers are high-order leftovers. For each object $\mathcal{O}_{\kappa,j_k}$ generating leftovers, its leftovers (objects of the next period) are named $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2k-1}$ and $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2k}$. For all $j_k \le m_\kappa$, we set the area of the two corresponding first-order leftovers as

$$A_{\kappa+1,m_{\kappa+1}+2k-1} = \gamma_{m_{\kappa+1}+2j_k-1} \quad \text{and} \quad A_{\kappa+1,m_{\kappa+1}+2k} = \gamma_{m_{\kappa+1}+2j_k}, \quad (27)$$

initialize their used area as

$$a_{\kappa+1,m_{\kappa+1}+2k-1} = a_{\kappa+1,m_{\kappa+1}+2k} = 0, \quad (28)$$

and set their ancestors (or origins) as the purchased object that generated them, i.e.

$$o_{\kappa+1,m_{\kappa+1}+2k-1} = o_{\kappa+1,m_{\kappa+1}+2k} = (\kappa, j_k). \quad (29)$$

For all $j_k > m_\kappa$, we make the two corresponding high-order leftovers inherit the ancestor of the leftover $\mathcal{O}_{\kappa,j_k}$ that is generating them, i.e.

$$o_{\kappa+1,m_{\kappa+1}+2k-1} = o_{\kappa+1,m_{\kappa+1}+2k} = o_{\kappa,j_k}. \quad (30)$$

(Note that the "ancestor" is a pair that saves the instant and the index of the purchasable object from which the leftover derives.) Finally, taking into account the items produced in the single-period problem that was just solved, we must update the used area of the first-order leftovers of all preceding periods. For each item $\mathcal{I}_{\kappa i}$ $(i = 1,\ldots,n_\kappa)$, we proceed as follows. Variables $v_{\kappa ij} \in \{0,1\}$ indicate to which object the item was assigned. By (7), only one of the $v_{\kappa ij}$ is equal to one and all the other are null. Let j be the index (between 1 and $\bar{m}_\kappa$) such that $v_{\kappa ij} = 1$. This means that the item was assigned to object $\mathcal{O}_{\kappa j}$. If $j \le m_\kappa$, then the object is a purchasable objects and there is nothing to be done. Otherwise, item $\mathcal{I}_{\kappa i}$ was produced from a leftover. So, we add its area, given by $w_{\kappa i} \times h_{\kappa i}$, to the used area of the ancestor $o_{\kappa j}$ of $\mathcal{O}_{\kappa j}$, i.e.

$$a_{o_{\kappa j}} \leftarrow a_{o_{\kappa j}} + w_{\kappa i} \times h_{\kappa i}. \quad (31)$$

(Note that $o_{\kappa j}$ is a pair of the form $o_{\kappa j} = ([o_{\kappa j}]_1, [o_{\kappa j}]_2)$. So, notation $a_{o_{\kappa j}}$ means $a_{([o_{\kappa j}]_1,[o_{\kappa j}]_2)}$.) At the end of the current ηth cycle, we are ready to compute the actual utilization rates of the first-order leftovers given by

$$f^{\eta}_{\kappa,2j-1} = \frac{a_{\kappa+1,m_{\kappa+1}+2j-1}}{A_{\kappa+1,m_{\kappa+1}+2j-1}} \quad \text{and} \quad f^{\eta}_{\kappa,2j} = \frac{a_{\kappa+1,m_{\kappa+1}+2j}}{A_{\kappa+1,m_{\kappa+1}+2j}} \quad (32)$$

Then, the δ's are updated as in (25). If (26) holds, the method stops. Otherwise, we update $\eta \leftarrow \eta+1$ and start a new cycle. The method also stops if in ten consecutive cycles the best solution found so far is not updated. Algorithm 1 summarizes the whole procedure.

4. Numerical experiments

In this section, we aim to evaluate the performance of the proposed forward-looking approach. The single-period models $\mathcal{M}(\kappa, \kappa+1)$ and $\mathcal{M}(\delta, \kappa, \kappa+1)$ were implemented in C/C++ using the ILOG Concert Technology. The myopic and the proposed forward-looking matheuristic approaches were also implemented in C/C++. Models and code are available at https://github.com/oberlan/bromro2. Code was compiled with g++ from gcc version 7.5.0 (GNU compiler collection) with the -O3 option enabled. Numerical experiments were conducted using a machine with Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz with 160 GB of RAM memory, and Ubuntu Server 20.04.4 operating system. Single-period instances within the myopic and the forward-looking approaches were solved using IBM ILOG CPLEX 22.1.0. A solution is reported as optimal by CPLEX when

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

or

$$\text{relative gap} = \frac{|\text{best feasible solution} - \text{best lower bound}|}{10^{-10} + |\text{best feasible solution}|} \le \varepsilon_{\text{rel}}, \quad (33)$$

where, by default, $\varepsilon_{\text{abs}} = 10^{-6}$ and $\varepsilon_{\text{rel}} = 10^{-4}$, and "best feasible solution" means the smallest value of the objective function related to a feasible solution generated by the method. The objective functions (3) and (24) of models $\mathcal{M}(\kappa, \kappa+1)$ and $\mathcal{M}(\delta, \kappa, \kappa+1)$, respectively, for $\kappa = p,\ldots,P-1$, assume large integer values at feasible points. Thus, a stopping criterion based on a relative error less than or equal to $\varepsilon_{\text{rel}} = 10^{-4}$ has 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 32,000, respectively; so the Branch & Bound tree is partially transferred to disk if memory is exhausted. All other parameters of the solver were used with their default values. This includes the deterministic parallel MIP optimizer to solve a mixed integer programming problem.

4.1. Parameters tuning

In a first set of experiments, we aim to analyze the behavior of the forward-looking approach for variations of its two parameters δ_{ini} and σ. Recall that $\delta_{\text{ini}} \in [0,1]$ corresponds to the initial value of the leftovers utilization fraction; while $\sigma \in (0,1)$ plays a role in the utilization fraction update rule in (25). Preliminary results, focused on avoiding premature terminations and thus obtaining good quality solutions, led us to set $\epsilon = 0.01$. In the numerical experiments of this section, we considered the twenty five instances with four periods introduced in Birgin et al. (2020), varying their leftovers "expiration date" parameter $\xi \in \{1,2,3,4\}$. All instances have up to 3 objects and up to 9 items per period. For completeness, tables describing each instance are given in Appendix. The experiments in Birgin et al. (2020) show that, when applied to these one hundred instances, CPLEX with a CPU time limit of two hours found an optimal solution in 91 cases. Therefore, we applied the forward-looking approach with all combinations of δ_{ini} and $\sigma \in \{0.5, 0.55, \ldots, 1.0\}$ to these 91 instances and computed the gap to the known optimal solution computed by CPLEX.

Fig. 7 (top) shows the average gap (over the 91 instances) for each combination of δ_{ini} and σ. The figure shows that best results are obtained for the combination $(\delta_{\text{ini}}, \sigma) = (0.9, 0.9)$. The graphic also shows that, as desired, small variations in the parameters produce smalls variations in the average results of the method. It should be noted that the number of cycles (or iterations) η that are performed

Algorithm 1: FORWARD-LOOKING APPROACH

Input: Let $\delta_{\text{ini}} \in (0,1)$, $\sigma \in (0,1)$, and $\epsilon > 0$ be given algorithmic parameters. Let p, P, m_κ, n_κ, $W_{\kappa,j}$, $H_{\kappa,j}$, $c_{\kappa,j}$, $w_{\kappa,i}$, $h_{\kappa,i}$ ($\kappa = p, \ldots, P-1$, $j = 1, \ldots, m_\kappa$, $i = 1, \ldots, n_\kappa$), d, $\bar{w}_i$, $\bar{h}_i$ ($i = 1, \ldots, d$), and ξ be given data describing an instance of the multi-period problem

Output: A solution to the multi-period problem $\mathcal{M}(p, P)$

$\eta \leftarrow 0$ // set the cycles' counter
// initialize the estimated utilization rates of the first-order leftovers of all periods
$\delta^{\eta}_{\kappa,2j-1} \leftarrow \delta_{\text{ini}}$ and $\delta^{\eta}_{\kappa,2j} \leftarrow \delta_{\text{ini}}$ ($\kappa = p, \ldots, P-1$, $j = 1, \ldots, m_\kappa$)
repeat
  // set first-period objects' number and dimensions (all of them are purchasable objects)
  Set $\bar{m}_p \leftarrow m_p$ and $\bar{W}_{pj} \leftarrow W_{pj}$ and $\bar{H}_{pj} \leftarrow H_{pj}$ ($j = 1, \ldots, m_p$)
  $e_{pj} \leftarrow \xi$ ($j = 1, \ldots, m_p$) // initialize their validity
  for $\kappa = p, \ldots, P-1$ **do**
    Solve $\mathcal{M}(\delta^{\eta}; \kappa, \kappa+1)$ // solve the κth single-period subproblem
    // for each item produced in the period
    for $i = 1, \ldots, n_\kappa$ **do**
      // set j as the index of the object $\mathcal{O}_{\kappa j}$ to which item $\mathcal{I}_{\kappa i}$ was assigned
      Let j be the only index for which $v_{\kappa ij} = 1$
      // if the object is a leftover
      if $j > m_\kappa$ **then**
        // add the item's area to the used area of the leftover's first-order ancestor
        According to (31), update the used area of the leftover's first-order ancestor
    Compute $\hat{m}_\kappa$ and $j_1 \le j_2 \le \cdots \le j_{\hat{m}_\kappa}$ such that $e_{\kappa,j_k} > 0$ // objects generating leftovers
    // process the leftovers to be used in forthcoming periods
    for $k = 1, \ldots, \hat{m}_\kappa$ **do**
      // leftovers of object $\mathcal{O}_{\kappa,j_k}$ are named $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2k-1}$ and $\mathcal{O}_{\kappa+1,m_{\kappa+1}+2k}$
      if $j_k \le m_\kappa$ **then**
        // leftovers are first-order (produced from a purchasable object)
        According to (27), (28), (29), for the two first-order leftovers of object $\mathcal{O}_{\kappa,j_k}$, compute their area, initialize with zero their used area, and set their common ancestor
      else
        // leftovers are high-order (leftovers of leftovers)
        According to (30), set their common ancestor
      // set the two leftovers' validity as their common ancestors' validity reduced by one
      Let $(\nu, \zeta) = o_{\kappa+1,m_{\kappa+1}+2k}$. Set $e_{\kappa+1,m_{\kappa+1}+2k-1} \leftarrow e_{\nu,\zeta} - 1$ and $e_{\kappa+1,m_{\kappa+1}+2k} \leftarrow e_{\nu,\zeta} - 1$
    // set next-period objects' number and dimensions (purchasable objects and leftovers)
    Set $\bar{m}_{\kappa+1} \leftarrow m_{\kappa+1} + 2\hat{m}_k$, $\bar{W}_{\kappa+1,j} \leftarrow W_{\kappa+1,j}$ and $\bar{H}_{\kappa+1,j} \leftarrow H_{\kappa+1,j}$ ($j = 1, \ldots, m_{\kappa+1}$), and, following constraint (11), set the dimensions $\bar{W}_{\kappa+1,j}$ and $\bar{H}_{\kappa+1,j}$ ($j = m_{\kappa+1}+1, \ldots, \bar{m}_{\kappa+1}$) as the dimensions of the corresponding leftovers
    $e_{\kappa+1,j} \leftarrow \xi$ ($j = 1, \ldots, m_{\kappa+1}$) // initialize the next-period purchasable objects' validity
  // compute the first-order leftovers' actual utilization rate of the cycle and update their estimated utilization rate
  for $\kappa = p, \ldots, P-1$ **do**
    for $j = 1, \ldots, m_\kappa$ **do**
      According to (32), (25), compute $f^{\eta}_{\kappa,2j-1}$ and $f^{\eta}_{\kappa,2j}$ and then compute $\delta^{\eta+1}_{\kappa,2j-1}$ and $\delta^{\eta+1}_{\kappa,2j}$
  Check the stopping criterion (26) and, if does not hold, update the cycles' counter $\eta \leftarrow \eta + 1$
until *stopping criterion* (26) *holds*
return the computed solution to the multi-period problem $\mathcal{M}(p, P)$ as the composition of the solutions to the single-period subproblems $\mathcal{M}(\delta^{\eta}; \kappa, \kappa+1)$ for $\kappa = p, \ldots, P-1$

until the satisfaction of the stopping rule (26) depends on δ_{ini} and σ. Fig. 7 (middle and bottom) displays the average number of cycles η and the average elapsed CPU time in seconds, as a function of δ_{ini} and σ. On the one hand, the CPU time has a low dependence on σ and, roughly speaking, is an increasing function of δ_{ini}. On the other hand, the number of cycles has a low dependence on δ_{ini} and increases as σ increases. Note that, when $\sigma = 1$, the rule (25) reduces to, at each cycle, discarding information of previous cycles and defining the utilization fraction as the actual utilization fraction of the cycle. In this case, the stopping rule (26) is satisfied if and only if the utilization rates of all objects are the same for two consecutive cycles. Fig. 7 shows that, actually, this phenomenon occur; but it produces a premature stopping with lower quality solutions. Anyway, regardless of the metrics related to computational cost, based on the quality of the solutions obtained, we selected $(\delta_{\text{ini}}, \sigma) = (0.9, 0.9)$ for the rest of the experiments.

Fig. 8 shows the data of Fig. 7 (top and middle) in a different way. Using bi-objective optimization concepts, it illustrates the Pareto frontier for the $(\delta_{\text{ini}}, \sigma)$ pairs using as conflicting objectives the computational effort (CPU time) and the result obtained (gap). The figure shows all pairs $(\delta_{\text{ini}}, \sigma)$ considered, with δ_{ini} and $\sigma \in \{0.50, 0.55, \ldots, 1.00\}$. Each point is represented by a small ball such that, the outer color corresponds to δ_{ini} and the inner color corresponds to σ. The figure clearly shows that, by choosing $(\delta_{\text{ini}}, \sigma) = (0.90, 0.90)$, we sacrifice performance and opt for the combination that delivers the best results with a high computational cost. The figure also clearly shows that, by sacrificing the quality of the solution found, it is still possible to find good quality solutions for much less time. For example, the pair

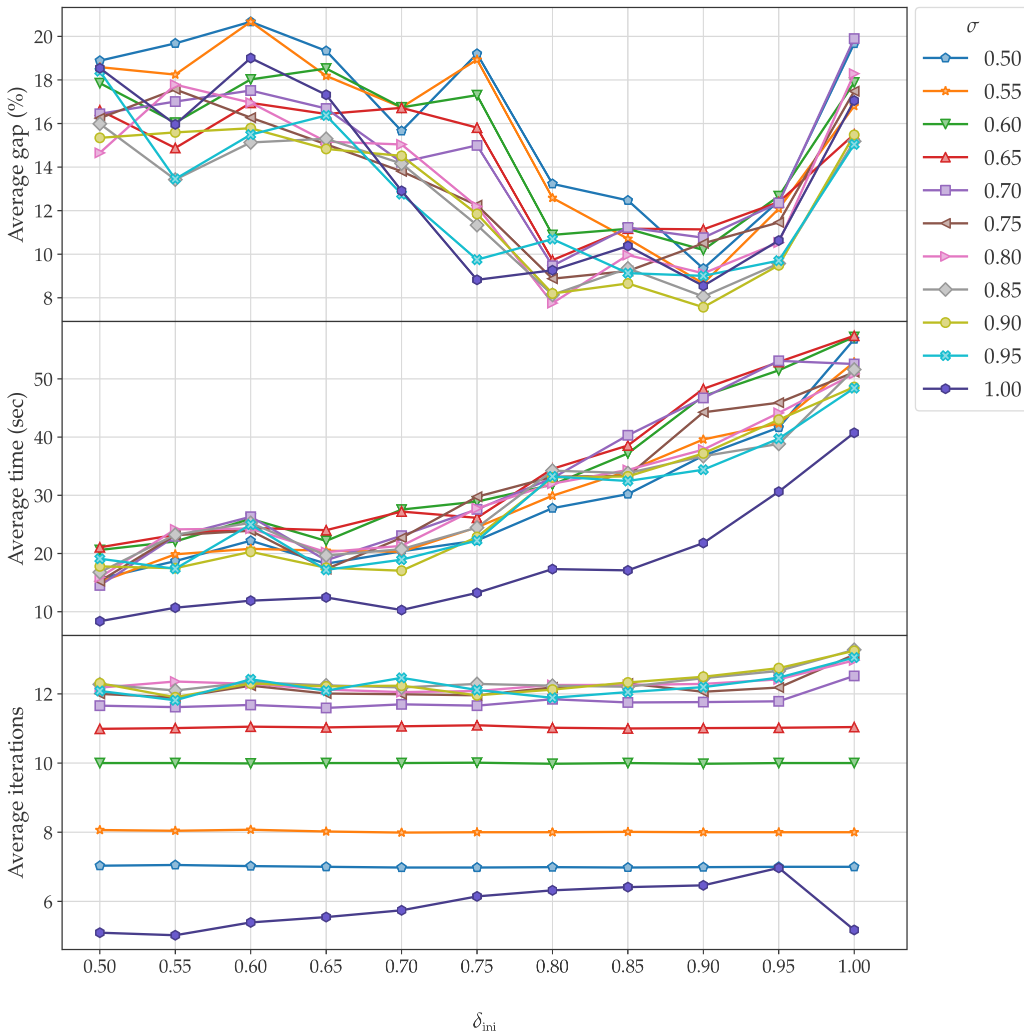

Fig. 7. Average gap (to optimal solution computed with CPLEX), CPU time (in seconds), and number of cycles of the forward-looking approach for variations of its parameters δ_{ini} and σ.

$(\delta_{\text{ini}}, \sigma) = (0.75, 1.00)$ manages to find solutions with very little loss of quality (the gap increases from 7.57% to 8.52%) with a two-thirds reduction in time. A four-fifths reduction in time can be obtained if solutions with a gap of 18.54% are satisfactory or if CPU time is a limiting factor.

4.2. Forward-looking versus myopic approach

In a second set of experiments, we compare the introduced forward-looking approach with $(\delta_{\text{ini}}, \sigma) = (0.9, 0.9)$ against the myopic approach, that only differs with the forward-looking approach in the objective function that is minimized in each subproblem. In this comparison, a new set of thirty instances with four, eight, and twelve periods, ten of each type, is considered. Instances were generated with the random generator introduced in Birgin et al. (2020). All instances, larger than the ones considered in Birgin et al. (2020), have up to 5 objects and up to 15 items per period. For the cases with four periods we consider instances with $\xi \in \{1, 2, 3, 4\}$ and for the cases with eight and twelve periods we consider $\xi \in \{1, 2, 3, 4, P\}$, totalizing 140 instances. Experiments that will be shown in the following sections show that CPLEX with a CPU time limit of two hours was able to find a guaranteed optimal solution in only 15 of the 140 instances. In order to allow reproducibility, a table describing each instance is given in Appendix. Table 1 shows the number of binary variables, continuous variables, and constraints of each instance. Note that instances with twelve periods and $\xi = P$ have around 400,000 binary variables, 300,000 continuous variables, and 4,000,000 constraints.

Tables 2–6 show the results. The tables show, for the myopic and the forward-looking approaches, the best objective function value found (i.e. the value of (3)), the corresponding cost of the purchased objects, the corresponding value of the leftovers at the final instant of the time horizon, and the CPU time in seconds. In addition, for the forward-looking approach, tables show the gap given by

$$100 \left(\frac{F_{\text{flook}} - F_{\text{myopic}}}{F_{\text{myopic}}} \right) \%, \tag{34}$$

where F_{flook} is the best objective function value found by the forward-looking approach and F_{myopic} is the best objective function value found by the myopic approach. It is important to notice that, by definition, the objective function (3) is dominated by the objects' cost (which is multiplied by an upper bound on the value of the leftovers at the last time instant); while the value of the leftovers at the last time instant plays a "tie-breaking role". Thus, a tiny gap may represent a situation where both methods have found a solution with the same cost of the objects but with a relevant difference in the value of the leftovers at instant P. Also note that Tables 2–6 do not include averages in the columns corresponding to the leftovers values. This is because, in the considered problem, the main goal is to find a solution that minimizes the overall cost of the objects and, among solutions with minimum costs of the objects, a solution that maximizes the value of the leftovers

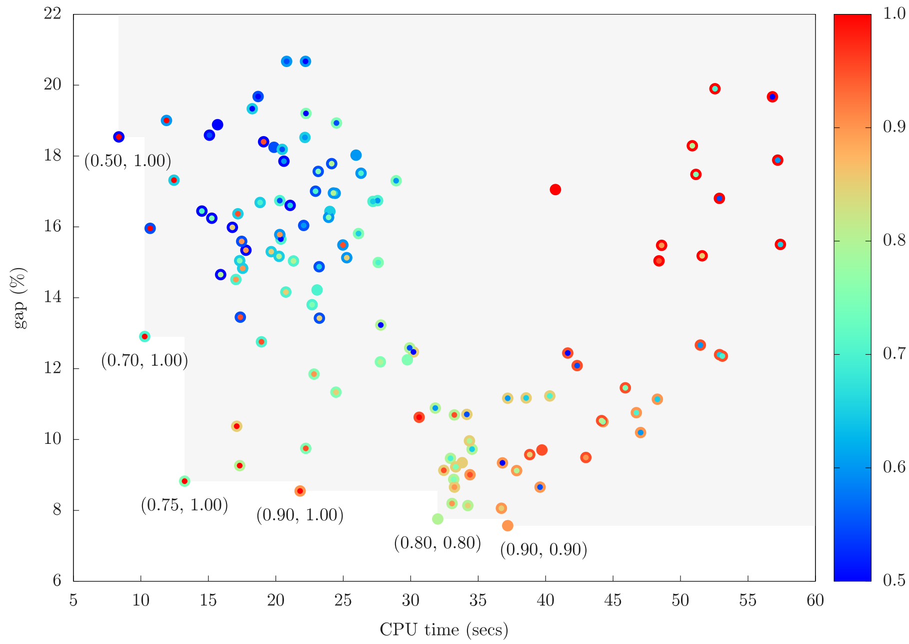

Fig. 8. Each pair $p = (\delta_{\mathrm{ini}}, \sigma)$ is associated with an average CPU time $t(p)$ and an average gap $g(p)$. We say that a pair p is *dominated* if there exists another pair q such (a) $t(q) < t(p)$ and $g(q) \le g(p)$ or (b) $t(q) \le t(p)$ and $g(q) < g(p)$. This image highlights the *non-dominated* pairs.

Table 1
Number of binary variables (BV), continuous variables (CV), and constraints (CO) of the thirty considered instances.

	Inst.	$\xi = 1$			$\xi = 2$			$\xi = 3$			$\xi = 4$			$\xi = P$		
		BV	CV	CO	BV	CV	CO	BV	CV	CO	BV	CV	CO	BV	CV	CO
4 periods	1	369	150	2,664	609	294	5,688	897	518	8,168	1,185	838	9,352			
	2	270	150	1,683	498	310	3,787	786	566	5,555	1,218	1,046	7,331			
	3	298	176	1,854	450	304	3,122	626	496	4,074	754	656	4,634			
	4	397	152	2,649	529	240	3,805	721	384	5,205	1,041	704	6,453	Since instances from 1 to 10 have $P = 4$ periods, the case $\xi = P$ coincides with the case $\xi = 4$.		
	5	487	150	3,752	695	254	6,932	951	430	9,396	1,335	910	11,076			
	6	290	202	1,809	546	402	3,845	898	754	5,757	1,042	914	6,349			
	7	572	214	4,443	844	358	8,667	1,164	630	11,683	1,308	790	12,275			
	8	503	154	3,328	675	282	5,456	979	426	11,560	1,235	746	12,680			
	9	318	196	2,044	538	380	3,672	706	556	4,520	1,138	1,036	6,296			
	10	345	162	2,072	525	290	3,584	749	434	5,784	1,069	754	7,032			
8 periods	11	1,028	444	9,014	1,848	868	19,982	3,368	1,668	40,422	5,672	2,820	70,806	28,904	21,764	265,142
	12	1,116	394	9,701	1,872	754	20,881	3,040	1,378	35,801	4,848	2,338	58,953	30,096	19,874	324,841
	13	593	362	3,824	1,105	722	8,004	1,889	1,298	14,092	3,281	2,418	22,780	20,625	16,818	113,308
	14	921	374	7,804	1,609	734	17,444	2,721	1,358	32,308	4,673	2,414	60,884	23,297	18,286	238,260
	15	986	390	8,311	1,702	742	17,911	2,982	1,430	33,255	5,334	2,710	62,487	25,910	17,558	228,343
	16	974	408	7,886	1,782	840	19,586	2,982	1,528	36,114	5,174	2,616	69,122	31,094	26,168	257,986
	17	1,251	394	10,836	2,071	714	26,772	3,455	1,386	50,388	5,631	2,282	91,972	27,359	16,362	432,452
	18	839	380	6,413	1,467	756	13,393	2,483	1,460	23,449	3,859	2,420	36,057	18,547	15,924	130,777
	19	1,020	400	8,012	1,660	720	16,656	2,780	1,296	31,432	4,620	2,320	53,288	22,956	17,488	202,888
	20	1,141	414	10,206	1,825	774	19,074	2,977	1,350	34,826	5,089	2,374	66,490	30,401	19,334	377,914
12 periods	21	1,184	514	8,941	2,056	978	19,957	3,728	1,842	42,077	6,784	3,442	82,925	343,904	246,834	3,855,917
	22	1,559	576	13,531	2,595	1,080	29,079	4,483	1,944	58,567	7,827	3,544	108,343	307,763	248,728	2,474,167
	23	1,158	530	8,965	2,066	1,050	19,405	3,794	1,994	40,397	6,626	3,594	73,149	326,178	295,370	2,276,765
	24	1,258	562	9,857	2,198	1,058	21,645	3,838	1,986	40,837	7,086	3,714	81,909	370,446	314,050	2,672,821
	25	1,443	584	12,671	2,403	1,096	25,275	4,283	2,104	50,827	7,387	3,928	88,299	359,931	319,320	2,927,211
	26	1,230	524	9,706	2,218	1,028	22,226	3,970	1,892	44,954	7,202	3,588	83,226	395,682	263,684	3,072,506
	27	1,452	558	11,777	2,480	1,054	26,525	4,472	2,030	56,773	7,928	3,790	108,821	482,392	405,326	4,270,261
	28	1,587	546	13,404	2,567	1,010	28,464	4,471	1,874	59,328	8,135	3,570	119,344	417,927	269,042	5,343,952
	29	1,488	656	12,636	2,596	1,224	27,628	4,588	2,264	54,436	8,300	4,152	106,004	480,652	339,576	6,202,740
	30	1,299	630	10,782	2,363	1,198	24,670	4,259	2,238	49,086	7,315	4,126	82,830	435,731	336,414	4,289,870

at instant P. Thus, it makes no sense to compare the value of the leftovers at instant P of solutions with different objects cost. It would be very easy to construct a solution with high objects cost and plenty of leftovers at the end of the considered time horizon. Given two solutions, the one with lower objects cost is better than the other; and in case the objects cost is identical, the one with the higher value of the leftovers at instant P is preferable. Solutions must be compared with this objective in mind; so the gaps must be examined carefully.

From what was recalled in the previous paragraph, by the definition of the problem, to win means to find a solution with strictly lower cost of the objects or with equal cost of the objects and strictly higher value of the leftovers at instant P. To tie means to find a solution with the same cost of the objects and the same value of the leftovers at instant P. If the method does not win or does not tie, then it loses. In Tables 2–6, values in bold correspond to the cases in which the method wins or ties. Table 7 summarizes the results. Each cell of the table is of the form

Table 2
Myopic approach versus forward-looking approach considering the scenario with smallest possible use of leftovers, i.e. $\xi = 1$.

	Inst.	Myopic approach				Forward-looking approach				
		Best objective function value	Objects cost	Leftovers value	CPU time	Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
4 periods	1	**314,108,050**	9,155	0	60.3	400,703,843	11,679	2,647	681.1	27.5688
	2	**187,422,365**	6,715	0	32.3	**187,422,365**	6,715	0	101.0	0.0000
	3	**340,487,089**	8,951	0	4.0	**340,487,089**	8,951	0	266.7	0.0000
	4	**309,586,584**	9,677	0	1.4	**309,586,584**	9,677	0	665.6	0.0000
	5	444,536,794	15,954	5,462	60.3	**182,258,424**	6,541	0	1,443.0	−59.0004
	6	236,240,392	6,246	2,066	0.3	**148,039,222**	3,914	0	157.4	−37.3353
	7	**607,520,858**	13,433	0	16.4	**607,520,858**	13,433	0	664.3	0.0000
	8	241,124,382	12,191	1,407	97.6	**191,042,687**	9,659	2,674	1,927.3	−20.7701
	9	**226,123,995**	4,757	0	0.8	**226,123,995**	4,757	0	257.8	0.0000
	10	**354,815,285**	10,884	3,115	9.2	**354,815,285**	10,884	3,115	287.4	0.0000
	Avg.	326,196,579	9,796		28.3	294,800,035	8,621		645.2	−8.9537
8 periods	11	1,550,317,180	16,165	3,310	180.4	**1,405,404,040**	14,654	2,484	3,136.5	−9.3473
	12	**1,625,463,920**	17,980	0	105.5	1,764,776,484	19,521	0	2,894.0	8.5706
	13	**1,102,076,378**	11,453	0	0.7	**1,102,076,378**	11,453	0	660.7	0.0000
	14	1,423,459,632	16,701	0	19.2	**1,360,217,488**	15,959	0	2,113.5	−4.4428
	15	1,156,701,480	15,396	0	160.0	**1,110,797,050**	14,785	0	3,136.7	−3.9686
	16	**1,037,649,354**	12,633	0	165.0	1,261,472,894	15,358	2,510	4,420.4	21.5702
	17	**1,236,188,630**	17,285	0	125.1	**1,236,188,630**	17,285	0	3,102.0	0.0000
	18	1,271,449,952	15,649	0	61.7	**1,234,400,864**	15,193	0	2,565.4	−2.9139
	19	**1,489,848,521**	17,883	2,092	126.1	1,559,998,475	18,725	0	1,814.8	4.7085
	20	**1,464,089,337**	17,855	2,808	63.9	1,555,845,650	18,974	3,376	2,084.8	6.2671
	Avg.	1,335,724,438	15,683		104.8	1,337,259,145	15,881		2,649.3	1.5752
12 periods	21	**2,905,035,501**	22,879	2,645	61.3	3,088,894,153	24,327	2,345	2,152.5	6.3290
	22	**2,526,326,584**	22,230	1,766	181.7	2,777,141,099	24,437	1,766	2,772.5	9.9280
	23	**2,554,150,135**	21,909	1,085	64.9	2,755,018,560	23,632	0	2,594.0	7.8644
	24	**2,745,092,742**	23,139	2,523	74.1	2,830,037,925	23,855	0	2,670.6	3.0944
	25	3,911,466,834	28,039	1,705	137.3	**3,423,912,544**	24,544	0	3,149.7	−12.4647
	26	**3,966,384,615**	27,042	735	124.9	4,586,379,445	31,269	1,130	2,847.6	15.6312
	27	**3,462,474,633**	26,709	0	241.0	3,674,042,217	28,341	0	3,034.6	6.1103
	28	3,134,068,124	28,791	4,972	161.8	**2,781,161,944**	25,549	0	3,625.9	−11.2603
	29	**2,682,280,094**	19,795	1,791	136.3	2,872,390,812	21,198	1,782	4,662.7	7.0877
	30	3,821,604,621	24,685	3,654	182.4	**3,730,110,699**	24,094	1,911	3,403.1	−2.3941
	Avg.	3,170,888,388	24,522		136.6	3,251,908,940	25,125		3,091.3	2.9926
	Avg.	1,610,936,469	16,739		88.5	1,635,275,590	16,645		2,109.8	−1.3056

"W/T/L G(%)", i.e. for each combination of number of periods $P \in \{4, 8, 12\}$ and parameter $\xi \in \{1, 2, 3, 4, P\}$ (comprising 10 instances), it displays the number of instances in which the forward-looking strategy wins, ties, and looses (with respect to the myopic approach), and the average gap given by (34). Figures in the table shows that, the larger the chance of taking advantage of leftovers (i.e. the larger ξ), the larger the number of victories and the larger the gap. This fact is graphically evidenced in Fig. 9. Clearly, the way to estimate the future impact of current decisions is heuristic in nature. This fact, associated with an instance in which there is little chance of using leftovers from previous periods (small ξ) occasionally leads the myopic method to obtain better results. This is an expected behavior that does not diminish the value of the proposed method. In the case $\xi = P$, which is the extreme case of the type of instances for which the method was developed, the forward looking approach find better solutions in all instances but one, with an average gap of, approximately, −15%.

4.3. Assessing the quality of small instances' solutions

In the previous section, numerical experiments made clear that the forward-looking approach outperforms the myopic approach; and the greater the possibility of saving raw material by employing leftovers (i.e. the larger the parameter ξ), the greater the advantage of the method. Since both methods differ in the looking-ahead objective function being minimized at each period, it is clear that this characteristic is well succeeded in that which it is intended to accomplish. On the other hand, we know nothing about how far from the optimal solution are the solutions that the method finds. In this section we perform an experiment comparing the solutions found by the forward-looking approach with the solutions found with CPLEX.

We consider in this experiment the ten instances with four periods and $\xi \in \{1, 2, 3, 4\}$. These problems, i.e. the corresponding multi-period models $\mathcal{M}(p, P)$, were solved with CPLEX, considering a time limit of two hours. The left-hand side of Table 8 shows the results. The table shows the ceiling of the best lower bound, the best objective function value found, the relative gap (33), and the CPU time in seconds. In addition, Since the value of the objective function (3) mixes the cost of the objects and the value of the leftovers at instant P and, thus, it is not very informative by itself, the table shows the cost of the objects and the value of the leftovers associated with each solution found. The right-hand side of the table gathers, from Tables 2–5, the results obtained by the forward-looking approach. In the right-hand side of the table, "gap(%)" represents the relative gap between the solutions found by both methods, computed as

$$100 \left(\frac{F_{\text{flook}} - F_{\text{cplex}}}{F_{\text{cplex}}} \right) \%, \tag{35}$$

where F_{flook} is the best objective function value found by the forward-looking approach and F_{cplex} is the best objective function value found by CPLEX. The table shows that, within the imposed CPU time limit, for $\xi = 1, 2, 3, 4$, CPLEX closed the gap in 7, 4, 4, and 0 instances (out of 10) respectively; while the average gap (35) between CPLEX and the forward-looking approach was 8.9%, 16.5%, 0.8%, and −7.6%. For the instances with $\xi = 1$, the forward-looking approach matched the solution found by CPLEX in 5 cases of which 4 are known to be optimal; and none solution was improved. For the instances with $\xi = 2$, the forward-looking approach matched 1 solution and improved other 2

Table 3
Myopic approach versus forward-looking approach considering the scenario with low use of leftovers, i.e. $\xi = 2$.

	Inst.	Myopic approach				Forward-looking approach				
		Best objective function value	Objects cost	Leftovers value	CPU time	Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
4 periods	1	**300,655,883**	8,763	2,647	0.9	**300,655,883**	8,763	2,647	1,190.8	0.0000
	2	**183,066,191**	6,559	2,058	0.8	187,421,443	6,715	922	557.7	2.3791
	3	340,482,337	8,951	4,752	63.4	**339,152,364**	8,916	3,360	481.9	−0.3906
	4	309,582,278	9,677	4,306	77.0	**277,209,196**	8,665	1,484	808.8	−10.4570
	5	274,293,216	9,844	0	120.1	**182,257,683**	6,541	741	1,446.4	−33.5537
	6	181,132,639	4,789	1,708	2.5	**179,167,551**	4,737	0	270.3	−1.0849
	7	**527,061,892**	11,654	1,912	134.2	634,518,412	14,030	2,368	1,633.8	20.3878
	8	**166,697,412**	8,428	0	38.0	**166,697,412**	8,428	0	913.2	0.0000
	9	226,123,365	4,757	630	7.6	**226,122,767**	4,757	1,228	458.4	−0.0003
	10	**284,400,266**	8,724	2,134	61.5	**284,400,266**	8,724	2,134	551.8	0.0000
	Avg.	279,349,548	8,215	5	50.6	277,760,298	8,028		831.3	−2.2720
8 periods	11	1,425,351,269	14,862	3,703	246.6	**1,383,057,725**	14,421	2,701	2,353.0	−2.9672
	12	1,492,298,384	16,507	444	301.6	**1,344,578,692**	14,873	0	3,650.4	−9.8988
	13	1,041,838,902	10,827	0	6.1	**741,805,859**	7,709	375	1,004.0	−28.7984
	14	**1,151,398,287**	13,509	801	55.3	1,253,506,060	14,707	964	2,746.1	8.8682
	15	1,190,883,867	15,851	1,763	138.1	**1,104,410,088**	14,700	912	3,563.9	−7.2613
	16	**1,037,649,024**	12,633	330	162.1	1,205,374,215	14,675	935	4,293.6	16.1640
	17	**1,137,778,191**	15,909	1,671	190.4	1,153,013,196	16,122	0	7,001.8	1.3390
	18	1,203,279,118	14,810	3,762	109.9	**1,025,673,954**	12,624	798	4,781.4	−14.7601
	19	**1,111,449,959**	13,341	2,092	194.0	1,183,682,688	14,208	0	4,432.4	6.4990
	20	**1,282,624,633**	15,642	3,725	127.1	1,386,189,384	16,905	3,711	4,830.0	8.0744
	Avg.	1,207,455,163	14,389		153.1	1,178,129,186	14,094		3,865.7	−2.2741
12 periods	21	**2,573,632,748**	20,269	3,258	137.9	2,737,557,920	21,560	1,520	4,216.8	6.3694
	22	**2,286,762,128**	20,122	2,562	269.2	2,415,976,613	21,259	2,442	3,767.1	5.6505
	23	2,324,372,040	19,938	0	234.3	**2,297,208,522**	19,705	378	6,005.3	−1.1686
	24	2,704,400,499	22,796	2,961	175.5	**2,623,019,850**	22,110	0	3,472.1	−3.0092
	25	3,755,224,476	26,919	2,943	190.0	**3,016,011,620**	21,620	0	5,557.5	−19.6849
	26	3,384,818,287	23,077	688	141.9	**2,863,535,290**	19,523	735	4,755.6	−15.4006
	27	**2,952,610,016**	22,776	2,296	309.1	3,082,376,653	23,777	2,296	7,569.5	4.3950
	28	2,991,904,474	27,485	2,686	237.9	**2,525,893,589**	23,204	1,035	3,854.2	−15.5757
	29	**2,369,810,439**	17,489	1,528	246.4	2,572,795,215	18,987	246	6,220.1	8.5654
	30	3,189,962,135	20,605	940	181.1	**2,682,942,759**	17,330	1,191	5,680.2	−15.8942
	Avg.	2,853,349,724	22,148		212.3	2,681,731,803	20,908		5,109.8	−4.5753
	Avg.	1,446,718,145	14,917		138.7	1,379,207,096	14,343		3,268.9	−3.0405

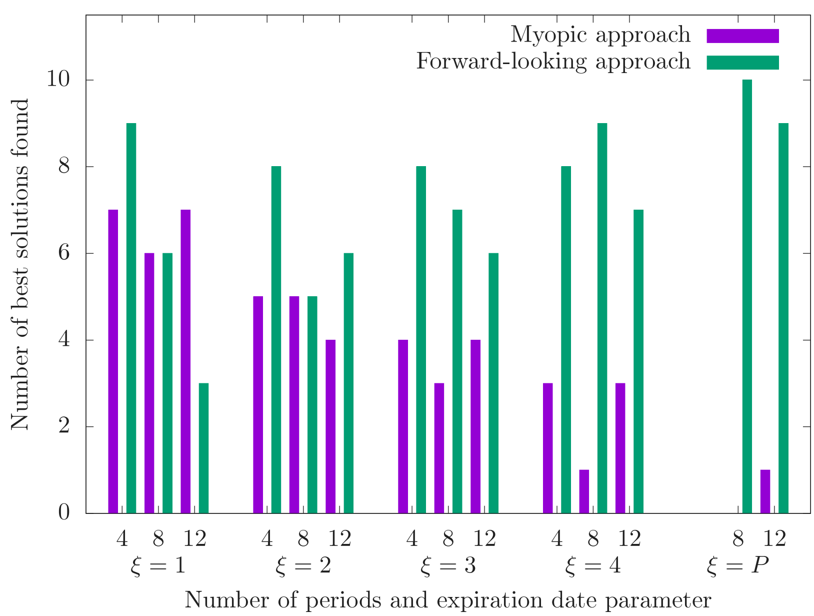

Fig. 9. Number of best solutions (including ties) found by the myopic and the forward-looking approaches as a function of the number of periods and the expiration date of the usable leftovers.

solutions. For the instances with $\xi = 3$, the forward-looking approach matched 2 solutions (known to be optimal) and improved other 3. For the instances with $\xi = 4$, the forward-looking approach improved 3 solutions found by CPLEX.

First of all, we should note that in this experiment we are considering instances with only four periods, which correspond to the smallest instances being considered in this work. Within this set, the cases in which CPLEX wins are concentrated in the instances with $\xi = 1, 2,$

Table 4
Myopic approach versus forward-looking approach considering the scenario with medium use of leftovers, i.e. $\xi = 3$.

Inst.		Myopic approach				Forward-looking approach				
		Best objective function value	Objects cost	Leftovers value	CPU time	Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
4 periods	1	**177,005,290**	5,159	0	60.6	277,051,660	8,075	1,590	922.0	56.5217
	2	183,066,047	6,559	2,202	2.7	**170,926,810**	6,124	154	355.7	−6.6311
	3	340,482,702	8,951	4,387	63.7	**205,638,144**	5,406	690	306.6	−39.6039
	4	309,582,332	9,677	4,252	122.0	**187,633,080**	5,865	0	808.2	−39.3915
	5	274,289,426	9,844	3,790	120.6	**182,257,479**	6,541	945	2,669.0	−33.5529
	6	181,132,281	4,789	2,066	2.4	**92,931,111**	2,457	0	160.1	−48.6943
	7	**352,310,540**	7,790	0	136.3	459,767,078	10,166	438	832.0	30.5005
	8	166,694,832	8,428	2,580	97.4	**143,119,673**	7,236	1,171	2,679.7	−14.1427
	9	**226,122,641**	4,757	1,354	9.1	**226,122,641**	4,757	1,354	347.7	0.0000
	10	**178,974,000**	5,490	0	65.4	**178,974,000**	5,490	0	314.8	0.0000
Avg.		238,966,009	7,144		68.0	212,442,168	6,212		939.6	−9.4994
8 periods	11	1,231,334,604	12,839	2,530	152.9	**1,165,832,300**	12,156	1,036	3,028.8	−5.3196
	12	1,555,759,878	17,209	2,558	303.9	**1,508,749,798**	16,689	2,558	3,621.5	−3.0217
	13	**920,593,767**	9,567	375	52.9	974,574,373	10,128	2,555	1,260.2	5.8637
	14	**1,019,203,389**	11,958	867	52.1	1,035,823,553	12,153	943	4,147.4	1.6307
	15	1,190,882,686	15,851	2,944	256.4	**1,048,738,758**	13,959	912	4,068.5	−11.9360
	16	1,210,381,894	14,736	3,674	144.5	**1,142,537,743**	13,910	1,837	3,256.4	−5.6052
	17	1,292,683,746	18,075	4,104	243.8	**1,126,765,124**	15,755	966	3,305.4	−12.8352
	18	**911,276,358**	11,216	1,210	175.0	1,025,673,954	12,624	798	4,028.9	12.5536
	19	1,111,449,683	13,341	2,368	208.1	**1,008,394,414**	12,104	1,930	4,102.8	−9.2721
	20	1,218,090,995	14,855	4,150	242.8	**1,066,068,178**	13,001	821	3,603.0	−12.4804
Avg.		1,166,165,700	13,965		183.2	1,110,315,820	13,248		3,442.3	−4.0422
12 periods	21	**2,289,085,834**	18,028	1,438	176.7	2,340,384,768	18,432	0	4,697.8	2.2410
	22	**2,073,905,839**	18,249	1,766	199.9	2,106,634,923	18,537	2,442	8,231.9	1.5781
	23	2,093,542,769	17,958	871	182.5	**2,032,571,215**	17,435	1,085	7,165.9	−2.9124
	24	2,704,399,467	22,796	3,993	194.1	**2,374,002,708**	20,011	2,277	2,767.2	−12.2170
	25	3,374,945,006	24,193	2,687	212.8	**2,891,294,796**	20,726	2,930	3,949.7	−14.3306
	26	**2,790,050,551**	19,022	1,299	194.4	2,929,979,126	19,976	674	3,723.2	5.0153
	27	**2,907,754,654**	22,430	3,256	320.6	3,085,098,730	23,798	2,596	5,565.1	6.0990
	28	2,947,923,296	27,081	6,040	342.2	**2,649,225,201**	24,337	3,271	3,963.2	−10.1325
	29	2,280,785,228	16,832	1,268	249.2	**2,174,821,959**	16,050	1,191	7,355.1	−4.6459
	30	2,677,059,585	17,292	1,395	245.4	**2,338,635,390**	15,106	0	3,055.4	−12.6416
Avg.		2,613,945,223	20,388		231.8	2,492,264,882	19,441		5,047.4	−4.1947
Avg.		1,339,692,311	13,832		161.0	1,271,674,290	12,967		3,143.1	−5.9121

which correspond to the smallest instances and to the instances in which there is little space to exploit leftovers. It is not expected the proposed method to be advantageous when the instance is so small that it can be solved optimally using CPLEX. On the other hand, the numbers show that (a) the proposed method finds solutions close to the optimal solutions when the optimal solutions are known and that, (b) even considering instances with as few as four periods, the larger the ξ, the greater the advantage of using the proposed method.

To corroborate the statements of the previous paragraph, we also experimented running CPLEX in the 20 most difficult instances, with 8 and 12 periods and $\xi \in \{4, P\}$. Table 9 shows the results. In 7 out of the 20 instances with $\xi = 4$, CPLEX failed to find a feasible solution; while it was able to find a feasible solution in the other 13 instances. However, the forward-looking approach found better results in all these 13 instances, with an average gap of −33.14%. Out of a total of 20 instances with $\xi = P$, CPLEX found a feasible solution in only 2 instances; and in these two cases the forward-looking approach found better solutions, with an average gap of −80.68%.

4.4. Discussion concerning computational cost, parallelism and further developments

This paper presents the first solution method reported in the literature to solve the multi-period cutting stock problem with usable leftovers introduced in Birgin et al. (2020). Being the first one, the numerical experiments of this work focused on getting good quality solutions, thinking of building a set of problems and solutions that could later be used as a benchmark for subsequent developments. This decision was evidenced in the choice of the parameters of the proposed method in Section 4.1: a small value for the parameter ϵ, which is directly related to the number of iterations of the method (the lower the ϵ the more iterations the method makes) and the choice of the parameters $(\delta_{\text{ini}}, \sigma)$, where we chose the combination that produced the best results at a high computational cost. However, the analysis of Fig. 8 showed that we can obtain results with essentially the same quality using one third of the computational effort and that if computational effort is a major constraint, we can still find good solutions in up to one fifth of the computational time.

The introduced method was compared in Section 4.2 with a myopic method and in Section 4.3 with the solutions found by an exact method with time limitation. The proposed method found better solutions than the myopic method, especially in the situations for which it was developed, i.e., instances in which there are many opportunities to take advantage of leftovers. However, the comparison with the myopic method can be considered unfair, considering that the proposed method uses more time than the myopic method. In this respect, it is worth remembering that we chose parameters that make the method expensive, and that solutions of similar quality could be found in a third of the time. Limiting the time of the proposed method to be close to the time used by the myopic method would be unreasonable as it would annihilate its potential advantages. The two methods are of the rolling horizon type, with the only difference being that the myopic method takes the best greedy decision at each instant while the proposed method has a vision of the future that is adjusted over time. Imposing on the proposed method a time limit equal to the time of the myopic method would be the same as allowing it to make a single iteration. In that single iteration, the initial estimate of the utilization of the leftovers would be used and the final solution would be totally

Table 5
Myopic approach versus forward-looking approach considering the scenario with high use of leftovers, i.e. $\xi = 4$.

	Inst.	Myopic approach				Forward-looking approach				
		Best objective function value	Objects cost	Leftovers value	CPU time	Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
4 periods	1	**177,003,277**	5,159	2,013	68.7	277,048,333	8,075	4,917	1,110.1	56.5216
	2	183,066,038	6,559	2,211	2.8	**170,926,618**	6,124	346	331.7	−6.6312
	3	340,482,702	8,951	4,387	63.6	**205,637,682**	5,406	1,152	304.6	−39.6041
	4	309,582,269	9,677	4,315	122.0	**187,632,447**	5,865	633	1,510.7	−39.3917
	5	274,288,892	9,844	4,324	180.1	**182,257,683**	6,541	741	1,513.6	−33.5527
	6	181,131,635	4,789	2,712	2.6	**92,930,802**	2,457	309	248.2	−48.6943
	7	**352,308,306**	7,790	2,234	194.4	459,763,070	10,166	4,446	1,421.0	30.5002
	8	166,694,832	8,428	2,580	97.7	**143,119,381**	7,236	1,463	3,001.7	−14.1429
	9	**226,122,426**	4,757	1,569	9.1	**226,122,426**	4,757	1,569	394.3	0.0000
	10	178,973,172	5,490	828	65.6	**178,972,994**	5,490	1,006	514.3	−0.0001
	Avg.	238,965,355	7,144		80.7	212,441,144	6,212		1,035.0	−9.4995
8 periods	11	1,173,599,085	12,237	2,637	80.9	**1,045,086,574**	10,897	1,108	3,000.9	−10.9503
	12	1,555,759,725	17,209	2,711	307.3	**1,283,103,972**	14,193	0	5,231.0	−17.5256
	13	1,006,137,497	10,456	1,559	61.0	**594,580,454**	6,179	0	1,498.7	−40.9047
	14	**1,019,202,506**	11,958	1,750	211.2	1,034,033,507	12,132	1,117	4,999.5	1.4552
	15	1,190,882,363	15,851	3,267	187.3	**901,935,429**	12,005	221	4,162.6	−24.2633
	16	1,210,381,894	14,736	3,674	202.0	**1,036,005,844**	12,613	750	3,397.7	−14.4067
	17	1,137,777,475	15,909	2,387	289.3	**1,083,925,806**	15,156	1,002	8,488.7	−4.7331
	18	1,203,278,753	14,810	4,127	188.6	**925,901,698**	11,396	510	4,619.3	−23.0518
	19	1,111,449,235	13,341	2,816	208.9	**1,026,389,411**	12,320	2,109	6,673.1	−7.6531
	20	1,282,623,747	15,642	4,611	309.6	**1,151,100,925**	14,038	1,037	4,336.0	−10.2542
	Avg.	1,189,109,228	14,215		204.6	1,008,206,362	12,093		4,640.7	−15.2287
12 periods	21	2,359,046,839	18,579	3,107	231.3	**2,248,327,065**	17,707	1,553	3,668.8	−4.6934
	22	**1,813,887,845**	15,961	0	219.4	1,876,619,885	16,513	0	5,023.8	3.4584
	23	2,061,483,504	17,683	636	228.8	**2,000,862,162**	17,163	378	6,223.4	−2.9407
	24	**2,301,874,270**	19,403	635	190.7	2,448,388,251	20,638	879	2,789.5	6.3650
	25	2,981,413,301	21,372	2,071	144.9	**2,704,226,259**	19,385	626	3,955.0	−9.2972
	26	2,929,977,951	19,976	1,849	255.4	**2,499,194,368**	17,039	957	4,805.9	−14.7026
	27	**2,698,521,316**	20,816	2,476	251.6	2,912,940,914	22,470	2,476	5,023.6	7.9458
	28	2,792,803,496	25,656	6,040	343.9	**2,569,542,877**	23,605	3,003	5,369.5	−7.9941
	29	2,491,626,343	18,388	2,821	194.7	**2,156,800,182**	15,917	1,069	6,318.9	−13.4381
	30	2,677,058,625	17,292	2,355	254.6	**2,225,775,255**	14,377	0	4,611.4	−16.8574
	Avg.	2,510,769,349	19,513		231.5	2,364,267,722	18,481		4,779.0	−5.2154
	Avg.	1,312,947,977	13,624		172.3	1,194,971,742	12,262		3,484.9	−9.9812

Table 6
Myopic approach versus forward-looking approach considering the scenario with unrestricted use of leftovers, i.e. $\xi = P$.

	Inst.	Myopic approach				Forward-looking approach				
		Best objective function value	Objects cost	Leftovers value	CPU time	Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
8 periods	11	1,215,891,809	12,678	4,459	192.3	**986,583,007**	10,287	2,015	3,695.6	−18.8593
	12	1,555,758,322	17,209	4,114	306.8	**1,328,123,427**	14,691	1,737	3,980.1	−14.6318
	13	773,366,591	8,037	1,771	68.8	**594,578,854**	6,179	1,600	1,202.0	−23.1181
	14	1,135,115,839	13,318	3,937	190.9	**900,474,708**	10,565	1,372	6,194.6	−20.6711
	15	1,190,882,178	15,851	3,452	155.4	**874,512,980**	11,640	220	3,263.3	−26.5660
	16	1,210,381,894	14,736	3,674	201.8	**909,595,796**	11,074	416	6,542.8	−24.8505
	17	1,137,777,262	15,909	2,600	289.3	**1,068,191,648**	14,936	1,200	6,391.5	−6.1159
	18	1,203,277,781	14,810	5,099	188.4	**926,144,408**	11,399	1,544	4,402.0	−23.0315
	19	1,111,448,881	13,341	3,170	269.4	**914,087,713**	10,972	579	7,378.1	−17.7571
	20	1,190,621,519	14,520	3,961	305.6	**880,914,146**	10,743	1,111	4,538.0	−26.0122
	Avg.	1,172,452,208	14,041		216.9	938,320,669	11,249		4,758.8	−20.1614
12 periods	21	1,983,206,578	15,619	328	176.2	**1,870,326,188**	14,730	832	5,020.4	−5.6918
	22	1,813,886,612	15,961	1,233	263.6	**1,765,473,725**	15,535	1,350	9,743.6	−2.6690
	23	**1,741,938,045**	14,942	315	274.9	1,787,753,330	15,335	970	5,817.6	2.6301
	24	2,301,871,943	19,403	2,962	188.4	**1,964,238,743**	16,557	952	4,650.0	−14.6678
	25	2,883,203,059	20,668	3,609	205.4	**2,428,432,517**	17,408	891	8,681.4	−15.7731
	26	2,790,048,502	19,022	3,348	195.0	**2,381,267,430**	16,235	1,195	6,079.0	−14.6514
	27	2,727,820,154	21,042	1,600	248.8	**2,410,469,370**	18,594	1,008	4,717.7	−11.6339
	28	2,303,933,956	21,165	3,284	317.5	**2,066,194,766**	18,981	970	8,722.4	−10.3188
	29	1,989,452,967	14,682	2,079	165.1	**1,955,577,574**	14,432	1,722	8,361.4	−1.7027
	30	2,677,060,181	17,292	799	244.3	**2,183,355,005**	14,103	940	4,166.0	−18.4421
	Avg.	2,321,242,200	17,980		227.9	2,081,308,865	16,191		6,595.9	−9.2920
	Avg.	1,746,847,204	16,010		222.4	1,509,814,767	13,720		5,677.4	−14.7267

determined by that arbitrary initial choice. (In particular, if $\delta_{\text{ini}} = 0$ were chosen, the solution would coincide with the solution of the myopic method). This discussion shows that the proposed method is by nature more time-consuming than the myopic method, since it was designed to make several iterations, each with a cost similar to the cost of the myopic method, iterations over which the method adjusts its

Table 7
Summary of the comparison between the myopic and the forward-looking approaches in the set of thirty instances with 4, 8, and 12 periods and $\xi \in \{1, 2, 3, 4, P\}$.

Periods	$\xi = 1$		$\xi = 2$		$\xi = 3$		$\xi = 4$		$\xi = P$	
	W/T/L	G(%)	W/T/L	G(%)	W/T/L	G(%)	W/T/L	G(%)	W/T/L	G(%)
4	3/6/1	−8.95	5/3/2	−2.27	6/2/2	−9.50	7/1/2	−9.50	–	–
8	4/2/4	1.58	5/0/5	−2.27	7/0/3	−4.04	9/0/1	−15.23	10/0/0	−20.16
12	3/0/7	2.99	6/0/4	−4.58	6/0/4	−4.19	7/0/3	−5.22	9/0/1	−9.29
Tot./Avg.	10/8/12	−1.31	16/3/11	−3.04	19/2/9	−5.91	23/1/6	−9.98	19/0/1	−14.73

Table 8
Comparison of the forward-looking approach solutions with the solutions found by CPLEX (two hours of CPU time limit) in the ten instances with four periods and $\xi \in \{1, 2, 3, 4\}$.

ξ	Inst.	CPLEX						Forward-looking approach				
		Ceiling of best lower bound	Best objective function value	Objects cost	Leftovers value	gap (%)	CPU time	Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
1	1	314,108,050	**314,108,050**	9,155	0	0.0000	0.2	400,703,843	11,679	2,647	681.1	27.5688
	2	183,065,998	**187,422,365**	6,715	0	2.3244	7,200.0	**187,422,365**	6,715	0	101.0	0.0000
	3	339,152,904	**339,152,904**	8,916	2,820	0.0000	0.8	340,487,089	8,951	0	266.7	0.3934
	4	309,586,584	**309,586,584**	9,677	0	0.0000	0.6	**309,586,584**	9,677	0	665.6	0.0000
	5	182,258,424	**182,258,424**	6,541	0	0.0000	504.0	**182,258,424**	6,541	0	1,443.0	0.0000
	6	148,039,222	**148,039,222**	3,914	0	0.0000	0.2	**148,039,222**	3,914	0	157.4	0.0000
	7	580,789,654	**580,790,380**	12,842	1,912	0.0001	7,200.0	607,520,858	13,433	0	664.3	4.6024
	8	80,065,392	**143,120,844**	7,236	0	44.0575	7,200.0	191,042,687	9,659	2,674	1,927.3	33.4835
	9	226,123,995	**226,123,995**	4,757	0	0.0000	0.3	**226,123,995**	4,757	0	257.8	0.0000
	10	288,510,000	**288,510,000**	8,850	0	0.0000	35.8	354,815,285	10,884	3,115	287.4	22.9820
	Avg.		271,911,277	7,860		4.6382	2,214.2	294,800,035	8.621		645.2	8.9030
2	1	277,053,250	**277,053,250**	8,075	0	0.0000	0.5	300,655,883	8,763	2,647	1,190.8	8.5192
	2	111,057,869	**125,208,746**	4,486	0	11.3018	7,200.0	187,421,443	6,715	922	557.7	49.6872
	3	205,638,834	**205,638,834**	5,406	0	0.0000	0.8	339,152,364	8,916	3,360	481.9	64.9262
	4	216,807,866	**277,209,196**	8,665	1,484	21.7891	7,200.0	**277,209,196**	8,665	1,484	808.8	0.0000
	5	162,302,164	235,866,350	8,465	2,410	31.1889	7,200.1	**182,257,683**	6,541	741	1,446.4	−22.7284
	6	136,049,331	**136,049,331**	3,597	0	0.0000	0.6	179,167,551	4,737	0	270.3	31.6931
	7	406,039,028	**491,516,168**	10,868	0	17.3905	7,200.0	634,518,412	14,030	2,368	1,633.8	29.0941
	8	80,063,257	186,631,657	9,436	2,987	57.1009	7,200.0	**166,697,412**	8,428	0	913.2	−10.6811
	9	226,118,364	**226,122,466**	4,757	1,529	0.0018	7,200.0	226,122,767	4,757	1,228	458.4	0.0001
	10	249,388,985	**249,388,985**	7,650	1,015	0.0000	495.9	284,400,266	8,724	2,134	551.8	14.0388
	Avg.		241,068,498	7,141		13.8773	4,369.8	277,760,298	8,028		831.3	16.4549
3	1	177,005,290	**177,005,290**	5,159	0	0.0000	0.9	277,051,660	8,075	1,590	922.0	56.5226
	2	111,054,694	**169,836,470**	6,085	1,965	34.6108	7,200.0	170,926,810	6,124	154	355.7	0.6409
	3	115,486,404	**205,637,468**	5,406	1,366	43.8398	7,200.0	205,638,144	5,406	690	306.6	0.0000
	4	127,232,184	309,582,103	9,677	4,481	58.9020	7,200.0	**187,633,080**	5,865	0	808.2	−39.3924
	5	53,610,336	203,212,152	7,293	0	73.6185	7,200.0	**182,257,479**	6,541	945	2,669.0	−10.3113
	6	92,931,111	**92,931,111**	2,457	0	0.0000	1.2	**92,931,111**	2,457	0	160.1	0.0000
	7	352,310,540	**352,310,540**	7,790	0	0.0000	14.1	459,767,078	10,166	438	832.0	30.5006
	8	36,551,592	203,245,012	10,276	3,992	82.0160	7,200.0	**143,119,673**	7,236	1,171	2,679.7	−29.5835
	9	226,118,158	**226,122,466**	4,757	1,529	0.0019	7,200.0	226,122,641	4,757	1,354	347.7	0.0000
	10	178,974,000	**178,974,000**	5,490	0	0.0000	57.0	**178,974,000**	5,490	0	314.8	0.0000
	Avg.		211,885,661	6,439		29.2989	4,327.3	212,442,168	6,212		939.6	0.8377
4	1	176,988,585	**177,003,343**	5,159	1,947	0.0083	7,200.0	277,048,333	8,075	4,917	1,110.1	56.5215
	2	111,047,720	**169,836,144**	6,085	2,291	34.6148	7,200.0	170,926,618	6,124	346	331.7	0.6421
	3	115,475,322	**205,637,138**	5,406	1,696	43.8451	7,200.0	205,637,682	5,406	1,152	304.6	0.0003
	4	127,219,873	314,860,429	9,842	4,835	59.5948	7,200.0	**187,632,447**	5,865	633	1,510.7	−40.4077
	5	53,602,745	274,288,791	9,844	4,425	80.4576	7,200.0	**182,257,683**	6,541	741	1,513.6	−33.5526
	6	92,926,077	**92,930,701**	2,457	410	0.0050	7,200.0	92,930,802	2,457	309	248.2	0.0001
	7	352,270,070	**459,762,829**	10,166	4,687	23.3800	7,200.0	459,763,070	10,166	4,446	1,421.0	0.0001
	8	36,542,003	347,683,703	17,579	11,338	89.4899	7,200.0	**143,119,381**	7,236	1,463	3,001.7	−58.8363
	9	226,116,418	**226,122,302**	4,757	1,693	0.0026	7,200.0	226,122,426	4,757	1,569	394.3	0.0001
	10	178,945,769	**178,972,776**	5,490	1,224	0.0151	7,200.0	178,972,994	5,490	1,006	514.3	0.0001
	Avg.		244,709,816	7,679		33.1413	7,200.0	212,441,144	6,212		1,035.0	−7.5632

vision of the future. Moreover, as shown in Tables 2–6 and Fig. 9, the proposed method outperforms the myopic approach in situations where there is plenty of room for leftover utilization (like in the cases $\xi = 4$ and $\xi = P$). In those situations, as detailed in Section 4.1 and reinforced in the previous paragraph, there is room to vary the parameters of the proposed method if the computational effort is a limiting factor.

The comparison with the solution found by an exact method like CPLEX with a time limit also deserves a little discussion. The imposed limit of two hours was arbitrary, related to the need to carry out a large number of experiments. However, experiments with limits of ten and twenty CPU hours were also performed, and no significant improvement was observed in the solutions found by CPLEX for the more relevant instances with 8 and 12 periods and $\xi \in \{4, P\}$ reported in Table 9. In fact, even with higher time limits, CPLEX was not even able to find feasible solutions in most of the problems, as already reported. These facts reinforce that modifying the comparison to impose the same time limit on the forward-looking approach that was imposed on CPLEX would also be an arbitrary choice, since this limit could be two, ten or twenty hours for CPLEX without any change in the quality of the solution found. (This fact is not rare and can be considered expected,

Table 9
Comparison of the forward-looking approach solutions with the solutions found by CPLEX (two hours of CPU time limit) in the twenty instances with eight and twelve periods and $\xi \in \{4, P\}$.

ξ	Inst.	CPLEX						Forward-looking approach				
		Ceiling of best lower bound	Best objective function value	Objects cost	Leftovers value	gap (%)	CPU time	Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
	11	473,195,099	1,181,466,014	12,319	0	59.9500	7,200.0	**1,045,086,574**	10,897	1,108	3,000.9	−11.5432
	12	Solution not found						**1,283,103,972**	14,193	0	5,231.0	–
	13	372,869,484	612,863,033	6,369	361	39.1600	7,200.0	**594,580,454**	6,179	0	1,498.7	−2.9831
	14	222,193,109	1,779,131,599	20,874	1,169	87.5100	7,200.0	**1,034,033,507**	12,132	1,117	4,999.5	−41.8799
	15	262,875,214	1,779,602,401	23,687	1,909	85.2300	7,200.0	**901,935,429**	12,005	221	4,162.6	−49.3181
	16	383,738,872	1,550,599,272	18,878	1,892	75.2500	7,200.0	**1,036,005,844**	12,613	750	3,397.7	−33.1867
	17	Solution not found						**1,083,925,806**	15,156	1,002	8,488.7	–
	18	451,919,637	1,202,550,436	14,801	1,212	62.4200	7,200.0	**925,901,698**	11,396	510	4,619.3	−23.0052
	19	618,233,892	1,642,891,463	19,720	1,457	62.3700	7,200.0	**1,026,389,411**	12,320	2,109	6,673.1	−37.5254
	20	Solution not found						**1,151,100,925**	14,038	1,037	4,336.0	–
4	Avg.		1,392,729,174	16,664		67.4129	7,200.0	1,008,206,362	12,093		4,640.8	−28.4917
	21	997,754,130	3,307,163,912	26,046	892	69.8300	7,200.0	**2,248,327,065**	17,707	1,553	3,668.8	−32.0165
	22	Solution not found						**1,876,619,885**	16,513	0	5,023.8	–
	23	1,113,795,638	2,764,226,978	23,711	1,402	59.7100	7,200.0	**2,000,862,162**	17,163	378	6,223.4	−27.6159
	24	992,737,680	3,085,338,077	26,007	2,368	67.8200	7,200.0	**2,448,388,251**	20,638	879	2,789.5	−20.6444
	25	664,024,760	3,857,201,153	27,650	1,497	82.7800	7,200.0	**2,704,226,259**	19,385	626	3,955.0	−29.8915
	26	930,197,235	4,649,597,500	31,700	0	79.9900	7,200.0	**2,499,194,368**	17,039	957	4,805.9	−46.2492
	27	Solution not found						**2,912,940,914**	22,470	2,476	5,023.6	–
	28	Solution not found						**2,569,542,877**	23,605	3,003	5,369.5	–
	29	Solution not found						**2,156,800,182**	15,917	1,069	6,318.9	–
	30	822,377,280	3,478,070,731	22,466	3,059	76.3600	7,200.0	**2,225,775,255**	14,377	0	4,611.4	−36.0055
	Avg.		3,523,599,725	26,263		72.7483	7,200.0	2,364,267,722	18,481		4,779.0	−32.0705
	11	Solution not found					7,200.0	**986,583,007**	10,287	2,015	3,695.6	–
	12	Solution not found					7,200.0	**1,328,123,427**	14,691	1,737	3,980.1	–
	13	191,251,115	4,109,235,104	42,704	0	95.3458	7,200.0	**594,578,854**	6,179	1,600	1,202.0	−85.5306
	14	Solution not found					7,200.0	**900,474,708**	10,565	1,372	6,194.6	–
	15	Solution not found					7,200.0	**874,512,980**	11,640	220	3,263.3	–
	16	Solution not found					7,200.0	**909,595,796**	11,074	416	6,542.8	–
	17	Solution not found					7,200.0	**1,068,191,648**	14,936	1,200	6,391.5	–
	18	303,252,735	3,831,863,430	47,163	35,994	92.0860	7,200.0	**926,144,408**	11,399	1,544	4,402.0	−75.8306
	19	Solution not found					7,200.0	**914,087,713**	10,972	579	7,378.1	–
	20	Solution not found					7,200.0	**880,914,146**	10,743	1,111	4,538.0	–
P	Avg.		3,970,549,267	44,934		93.7159	7,200.0	938,320,669	11,249		4,758.8	−80.6806
	21	Solution not found					7,200.0	**1,870,326,188**	14,730	832	5,020.4	–
	22	Solution not found					7,200.0	**1,765,473,725**	15,535	1,350	9,743.6	–
	23	Solution not found					7,200.0	**1,787,753,330**	15,335	970	5,817.6	–
	24	Solution not found					7,200.0	**1,964,238,743**	16,557	952	4,650.0	–
	25	Solution not found					7,200.0	**2,428,432,517**	17,408	891	8,681.4	–
	26	Solution not found					7,200.0	**2,381,267,430**	16,235	1,195	6,079.0	–
	27	Solution not found					7,200.0	**2,410,469,370**	18,594	1,008	4,717.7	–
	28	Solution not found					7,200.0	**2,066,194,766**	18,981	970	8,722.4	–
	29	Solution not found					7,200.0	**1,955,577,574**	14,432	1,722	8,361.4	–
	30	Solution not found					7,200.0	**2,183,355,005**	14,103	940	4,166.0	–
	Avg.		–	–	–	–	7,200.0	2,081,308,865	16,191		6,596.0	–

since as the bottom-right of Table 1 shows we are dealing with instances with a huge number of real and binary variables and constraints.)

In the present work, the implementation of the myopic method, the implementation of the proposed method, and the solution of the integrated multi-period problem lie in the use of CPLEX. By default, CPLEX uses a deterministic type of parallelism, which impacts all presented experiments in a similar way. For this reason, the use of parallelism for the resolution of the models, whether for a single period or for the multiperiod problem as a whole, was not highlighted. Naturally, any improvement in the parallelization of integer programming models would positively impact all tested methods. In particular, in practice, the considered (most updated) version of CPLEX has an opportunistic mode of parallelization that promises better results at the price of losing the determinism of the results. Apart from the possibility of parallelizing the resolution of the single period models, the rolling horizon methods such as the myopic method and the forward-looking proposed method are sequential in nature, since each period uses the solution of the previous period as input data.

The present work demonstrated that a forward-looking method has the potential to address large instances of the considered problem and find better quality solutions than a myopic method. The drawback of the proposed method, which makes it expensive and prevents it from addressing even larger instances, is the use of an exact method to solve the subproblems of a single period. For this reason, the next development should be to develop a heuristic method for the single-period cutting problem with usable leftovers introduced in Andrade et al. (2014). The instances and their solutions presented in the present work will serve as a set of tests for that and other future developments.

5. Concluding remarks

This work contributes to the literature on two-dimensional cutting stock problems with usable leftovers, which is very limited. A forward-looking approach for the multi-period two-dimensional non-guillotine cutting stock problem with usable leftovers, proposed in Birgin et al. (2020), was introduced, this being the first method reported in the literature to address this problem. The method solves a sequence of single-period subproblems and differs with a myopic approach in the objective function being minimized. On the one hand, the myopic approach greedily minimizes the cost of the raw material that must be purchased to produce the orders of the period. On the other hand, the forward-looking approach takes into consideration the future impact of the decisions of the period. This looking-head feature allows the method to suggest the purchase of some extra raw material whose leftovers are expected to be used in future periods, resulting in a lower overall cost. Numerical experiments shown the efficiency and effectiveness of the method. In summary, the proposed approach greatly improves the solution found with a commercial solver or with a myopic approach in problems with a reasonable number of periods in which usable leftovers can be used over several periods after they have been generated, i.e. a scenario in which leftovers can play a relevant role.

On the one hand, the proposed method can be applied to instances with a large number of periods. On the other hand, solving the

Table A.10
Summary of notations used throughout the article to define the considered problem and the introduced forward-looking approach.

Problem data	
p	initial instant of the considered horizon (zero when omitted)
P	final instant of the considered horizon
ξ	leftovers' expiration date
m_s	number of purchasable objects available at instant s
$\mathcal{O}_{sj}$	name of the jth purchasable object at instant s
W_{sj}	width of object $\mathcal{O}_{sj}$
H_{sj}	height of object $\mathcal{O}_{sj}$
c_{sj}	cost per unit of area of object $\mathcal{O}_{sj}$
n_s	number of items that must be produced along the period $[s, s+1)$
$\mathcal{I}_{si}$	name of the ith item that must be produced along the period $[s, s+1)$
w_{si}	width of item $\mathcal{I}_{si}$
h_{si}	height of item $\mathcal{I}_{si}$
d	number of items in the catalogue
$\bar{\mathcal{I}}_i$	name of the ith item in the catalogue
$\bar{w}_i$	width of ith item in the catalogue
$\bar{h}_i$	height of ith item in the catalogue
Parameters of the forward-looking approach	
δ_{ini}	initial utilization rate of the leftovers
σ	constant used to update the leftovers utilization rates
ϵ	stopping criterion tolerance

single-period subproblems exactly, even using parallelism, limits the applicability to instances with large single-period subproblems. Then, devising a heuristic method for the single-period problem would have an immediate impact on methods for solving the multi-period problem. That will be a subject of future work. In another line of research, the problem introduced in Birgin et al. (2020) and for which a method was developed in the present work, could be modified to take into account situations that sometimes arise in practice. For example, the problem could be modified to allow the anticipated production of items included in future period orders. In this case, storage costs and production capacity limits for each period could be considered. In addition, in the case of isotropic materials, ninety-degree rotations of the items could also be contemplated, as for example considered in Ayadia et al. (2017).

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ernesto G. Birgin reports financial support was provided by State of Sao Paulo Research Foundation. Oberlan C. Romao reports financial support was provided by Foundation for Research Support of Espírito Santo. Debora P. Ronconi reports financial support was provided by State of Sao Paulo Research Foundation. Ernesto G. Birgin reports financial support was provided by National Council for Scientific and Technological Development. Debora P. Ronconi reports financial support was provided by National Council for Scientific and Technological Development.

Data availability

Data will be made available on request.

Acknowledgments

This work has been partially supported by FAPESP (grants 2013/07375-0, 2016/01860-1, and 2018/24293-0), FAPES (grant 116/2019), and CNPq (grants 146110/2013-7, 311536/2020-4 and 302682/2019-8). The authors would like to thank the referees of this work, whose contributions helped to improve its quality. In particular, the authors are grateful to the referee who suggested the inclusion of Fig. 8, which helped to clearly present various aspects of the method introduced.

Appendix

Table A.10 summarizes the notations used throughout the article to define the considered problem and the introduced forward-looking approach. Table A.11 describes in detail the twenty five instances with four periods introduced in Birgin et al. (2020) and considered in the parameters tuning Section 4.1 of the present work. In addition, Table A.12 shows the number of binary variables, continuous variables, and constraints of each instance of Table A.11 when $\xi \in \{1,2,3,4\}$. Table A.13 describes in detail the thirty instances with four, eight, and twelve periods introduced in the present work. Instances were generated with the random instances generator introduced in Birgin et al. (2020) and available at https://github.com/oberlan/bromro2. The number of binary variables, continuous variables, and constraints of each instance of Table A.13, for $\xi \in \{1,2,3,4,P\}$ was given in Table 1. In Tables A.11 and A.13, 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. Column d represents the number of items in the catalogue, that are the ones whose dimensions are underlined in the tables.

Table A.11
Description of the twenty five instances with four periods taken from Birgin et al. (2020) and considered in the parameter tuning Section 4.1 of the present work.

Inst.	P	s	Objects $\mathcal{O}_{sj}$		Items $\mathcal{I}_{si}$		
			m_s	$W_{sj} \times H_{sj}$	n_s	d	$w_{si} \times h_{si}$
1	3	0	3	21 × 17,19 × 19,24 × 13	2	1	10 × 11,9 × 11
		1	1	10 × 16	3		7 × 6,7 × 5,7 × 4
		2	1	10 × 12	2		2(6 × 3)
2	4	0	2	14 × 8,16 × 6	3	2	3 × 7,6 × 8,4 × 8
		1	1	15 × 10	3		5 × 3, 2(2 × 5)
		2	1	20 × 15	2		5 × 3, 3 × 2
		3	1	15 × 10	2		2(2 × 3)
3	4	0	2	15 × 6,15 × 5	3	2	2(1 × 6), 10 × 6
		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, 2 × 2, 2(5 × 2)

(continued on next page)

Table A.11 (*continued*).

Inst.	P	s	Objects $\mathcal{O}_{sj}$		Items $\mathcal{I}_{si}$		
			m_s	$W_{sj} \times H_{sj}$	n_s	d	$w_{si} \times h_{si}$
4	4	0	2	13 × 8,12 × 6	5	2	1 × 5,2 × 5,1 × 4,1 × 3,3 × 2
		1	3	10 × 8,12 × 10,15 × 10	3		3 × 7,2 × 3,2 × 4
		2	1	8 × 4	2		10 × 1, 1 × 3
		3	0		3		3 × 1, 3 × 3,4 × 4
5	4	0	2	10 × 4,13 × 8	4	1	2(1 × 5), 2 × 5,3 × 5
		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		1 × 2, 5 × 4,4 × 2
6	4	0	2	22 × 17,14 × 30	5	4	3(2 × 11), 2(5 × 5)
		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(3 × 3), 4 × 2, 2(7 × 1), 11 × 1
7	4	0	2	(10 × 12) 2, 12 × 10	3	1	5 × 4,8 × 2, 2 × 2
		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	1	5 × 4,8 × 2, 2 × 2
		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	2	3 × 7,8 × 2,10 × 1,5 × 4,2 × 9, 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, 4 × 1
		3	0		4		8 × 4,4 × 2,3 × 7,6 × 2
10	4	0	2	14 × 21,19 × 19	7	1	2(11 × 3), 3(2 × 11), 2(5 × 5)
		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(2 × 1)
11	4	0	2	30 × 10,23 × 16	1	2	6 × 6
		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	1	2(11 × 6)
		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	2(6 × 6)
		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	1	2(3 × 12), 2(15 × 10)
		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	2(11 × 13)
		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	3	2(2 × 4), 2(6 × 16)
		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	2(6 × 2), 5 × 9
		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	1	2 × 11
		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

(*continued on next page*)

Table A.11 (*continued*).

Inst.	P	s	Objects $\mathcal{O}_{sj}$		Items $\mathcal{I}_{si}$		
			m_s	$W_{sj} \times H_{sj}$	n_s	d	$w_{si} \times h_{si}$
19	4	0	2	14 × 14,39 × 11	2	2	11 × 6,8 × 5
		1	1	15 × 23	3		2(6 × 10), $\underline{2 \times 10}$
		2	1	39 × 14	3		2(5 × 5), 7 × 2
		3	1	36 × 11	2		$\underline{3 \times 1}$, 3 × 2
20	4	0	1	27 × 24	3	4	$\underline{4 \times 6}$, 2($\underline{10 \times 2}$)
		1	2	35 × 27,27 × 11	1		14 × 5
		2	2	23 × 30,17 × 13	3		2(6 × 8), $\underline{5 \times 5}$
		3	1	24 × 34	2		2($\underline{3 \times 7}$)
21	4	0	3	10 × 17,26 × 15,12 × 11	1	1	$\underline{2 \times 1}$
		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	1	5 × 9,8 × 6, 2($\underline{2 \times 4}$)
		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	2(4 × 5), 2(1 × 10), 2(4 × 3)
		1	3	18 × 20,25 × 13,21 × 16	2		2(2 × 5)
		2	2	12 × 24,14 × 16	5		2($\underline{2 \times 2}$), 5 × 9, 2(6 × 2)
		3	1	14 × 27	4		3 × 6, 2(4 × 6), $\underline{1 \times 4}$
24	4	0	1	21 × 21	5	3	$\underline{4 \times 2}$, 2(3 × 9), 2(8 × 3)
		1	2	19 × 30,23 × 12	3		$\underline{2 \times 6}$, 8 × 5,5 × 4
		2	2	21 × 28,24 × 11	1		10 × 2
		3	1	29 × 16	2		2($\underline{3 \times 5}$)
25	4	0	3	22 × 28,30 × 25,19 × 22	2	2	2(6 × 5)
		1	2	22 × 22,12 × 22	4		2(4 × 8), 2($\underline{2 \times 3}$)
		2	1	22 × 11	4		3 × 3, $\underline{3 \times 1}$, 2(8 × 1)
		3	2	23 × 19,12 × 23	4		4 × 9,4 × 8, 2(7 × 9)

Table A.12
Number of binary variables (BV), continuous variables (CV), and constraints (CO) of the twenty five instances with four periods taken from Birgin et al. (2020) and considered in the parameter tuning Section 4.1 of the present work.

Inst.	$\xi = 1$			$\xi = 2$			$\xi = 3$			$\xi = 4$		
	BV	CV	CO	BV	CV	CO	BV	CV	CO	BV	CV	CO
1	81	82	410	153	162	802	297	354	1,498	297	354	1,498
2	89	88	423	165	168	823	285	296	1,439	509	552	2,431
3	115	92	547	211	172	1,227	403	300	2,947	659	556	4,003
4	121	98	588	237	206	1,292	461	438	2,508	653	662	3,404
5	127	108	651	267	256	1,507	427	432	2,467	587	656	3,299
6	276	168	1,704	468	296	3,448	772	488	6,392	1,060	744	7,512
7	61	80	275	121	160	583	217	288	1,071	409	544	1,999
8	61	80	275	121	160	583	217	288	1,071	409	544	1,999
9	204	112	1,392	360	208	2,568	672	496	4,488	1,008	880	5,976
10	373	132	2,171	521	212	4,231	713	340	6,351	905	596	7,279
11	108	100	546	224	212	1,202	360	340	1,994	584	596	2,986
12	85	106	382	161	190	786	281	326	1,410	505	614	2,434
13	87	114	374	167	214	766	343	422	1,566	599	710	2,654
14	106	86	482	174	154	898	262	258	1,338	374	402	1,850
15	108	94	514	212	178	1,238	372	314	2,294	628	602	3,382
16	106	102	494	206	202	1,006	374	370	1,846	806	802	3,574
17	113	104	513	181	172	877	285	276	1,421	413	420	1,965
18	143	136	735	275	272	1,491	483	520	2,531	595	664	3,043
19	92	90	440	180	174	940	308	310	1,580	564	598	2,668
20	101	100	466	233	220	1,090	425	396	1,882	585	540	2,490
21	87	114	402	211	274	1,034	355	434	1,818	643	818	3,210
22	112	128	510	224	272	1,078	336	400	1,678	528	656	2,606
23	224	150	1,301	436	310	2,953	748	598	4,873	1,084	982	6,361
24	109	102	460	217	214	972	377	374	1,684	505	502	2,212
25	180	140	996	328	252	2,100	584	476	3,772	920	860	5,260

Table A.13
Description of the considered thirty instances with four, eight, and twelve periods.

Inst.	P	s	Objects $\mathcal{O}_{sj}$		Items $\mathcal{I}_{si}$		
			m_s	$W_{sj} \times H_{sj}$	n_s	d	$w_{si} \times h_{si}$
1	4	0	2	77 × 100,67 × 77	4	2	2(6 × 5), 2(9 × 6)
		1	2	81 × 36,95 × 33	6		8 × 11, 2(15 × 6), 3(18 × 14)
		2	2	54 × 74,78 × 100	10		3(6 × 8), 3(7 × 9), 2(17 × 13), 2(13 × 8)
		3	1	53 × 68	7		3(10 × 5), 5 × 6, 18 × 15, 2(16 × 14)
2	4	0	3	49 × 82,34 × 70,57 × 76	6	2	2(7 × 5), 19 × 15, 3(17 × 15)
		1	2	39 × 54,39 × 41	4		17 × 20, 2(9 × 20), 20 × 17
		2	2	38 × 72,85 × 96	7		10 × 10, 3(14 × 8), 18 × 20, 2(6 × 18)
		3	1	43 × 60	4		14 × 8, 3(18 × 7)
3	4	0	1	69 × 44	4	1	15 × 6,14 × 8, 2(8 × 11)
		1	2	30 × 79,39 × 92	6		3(8 × 17), 3(18 × 17)
		2	2	83 × 89,65 × 91	8		13 × 11, 3(8 × 5), 2(9 × 14), 2(18 × 17)
		3	3	96 × 73,54 × 65,95 × 55	4		14 × 14, 2(10 × 15), 12 × 13
4	4	0	2	41 × 97,85 × 69	4	3	14 × 12, 2(18 × 8), 19 × 15
		1	1	90 × 95	13		3(14 × 10), 3(8 × 10), 2(19 × 12), 3(17 × 6), 2(17 × 9)
		2	1	75 × 76	6		18 × 12, 5 × 20, 2(15 × 20), 2(9 × 11)
		3	2	80 × 35,85 × 60	5		19 × 13, 3(16 × 14), 12 × 18
5	4	0	3	91 × 59,52 × 37,40 × 66	4	1	2(6 × 5), 2(19 × 14)
		1	1	88 × 90	13		2(20 × 9), 3(7 × 7), 2(7 × 15), 3(19 × 8), 3(11 × 16)
		2	1	83 × 47	10		3(20 × 8), 2(20 × 9), 3(14 × 18), 2(17 × 17)
		3	1	65 × 94	6		3(7 × 8), 3(17 × 9)
6	4	0	1	63 × 39	3	2	2(5 × 8), 12 × 7
		1	4	81 × 87, 2(38 × 30), 81 × 54	5		2(14 × 18), 3(7 × 19)
		2	3	83 × 91,47 × 31,52 × 71	3		16 × 6,16 × 9,7 × 11
		3	3	53 × 56,44 × 53,37 × 99	6		3(11 × 5), 14 × 19, 2(6 × 12)
7	4	0	1	82 × 95	7	2	12 × 17,10 × 5,9 × 17, 3(6 × 18), 12 × 20
		1	3	57 × 54, 2(33 × 36)	8		3(20 × 17), 2(11 × 8), 2(15 × 14), 18 × 5
		2	2	95 × 67,99 × 57	9		2(10 × 17), 5 × 8, 3(6 × 6), 3(14 × 9)
		3	3	42 × 92,88 × 100,85 × 86	11		15 × 15, 2(16 × 10), 2(6 × 5), 3(16 × 12), 3(12 × 17)
8	4	0	2	2(56 × 33)	10	1	3(13 × 17), 2(17 × 7), 17 × 10,7 × 13, 3(15 × 10)
		1	1	70 × 94	8		12 × 8, 2(9 × 7), 18 × 5, 3(14 × 13), 6 × 9
		2	2	55 × 40,60 × 59	4		3(16 × 9), 11 × 14
		3	1	71 × 53	13		3(16 × 19), 2(5 × 5), 2(18 × 6), 3(11 × 14), 3(12 × 18)
9	4	0	3	66 × 99,93 × 54,30 × 74	4	2	3(5 × 16), 11 × 16
		1	1	56 × 93	8		3(14 × 12), 14 × 10, 3(10 × 7), 19 × 10
		2	3	67 × 68,43 × 59,93 × 74	6		2(18 × 10), 13 × 17, 3(19 × 7)
		3	3	93 × 92,86 × 53,43 × 34	2		14 × 20,12 × 9
10	4	0	2	78 × 95,61 × 90	7	3	2(9 × 19), 2(12 × 6), 3(6 × 12)
		1	1	62 × 79	7		3(20 × 15), 3(15 × 7), 16 × 18
		2	2	36 × 60,35 × 96	6		2(16 × 16), 7 × 17, 3(9 × 8)
		3	2	84 × 72,33 × 98	7		2(11 × 5), 3(7 × 17), 20 × 16,19 × 12
11	8	0	3	61 × 85,37 × 95,84 × 46	4	2	16 × 20, 3(5 × 6)
		1	3	72 × 55,62 × 41,35 × 33	6		3(8 × 5), 8 × 17, 2(14 × 5)
		2	3	90 × 68,47 × 44,52 × 63	3		2(14 × 16), 14 × 17
		3	4	2(39 × 56), 81 × 81,61 × 44	10		2(19 × 19), 3(7 × 15), 2(16 × 15), 3(18 × 9)
		4	2	54 × 97,40 × 86	7		3(17 × 7), 13 × 6, 3(10 × 6)
		5	4	2(33 × 43), 93 × 77,84 × 70	9		3(16 × 16), 3(10 × 11), 3(14 × 11)
		6	3	41 × 74,86 × 91,62 × 30	8		3(19 × 8), 3(8 × 9), 2(7 × 6)
		7	3	100 × 37,69 × 65,83 × 62	7		2(13 × 18), 7 × 8,13 × 12, 2(12 × 7), 14 × 18
12	8	0	3	68 × 37,70 × 43,97 × 52	7	3	20 × 14,14 × 10,20 × 15, 3(17 × 19), 7 × 13
		1	3	88 × 39,89 × 35,55 × 79	8		3(7 × 17), 3(15 × 11), 10 × 12,20 × 10
		2	2	66 × 77,58 × 88	11		18 × 9, 3(10 × 20), 2(18 × 5), 2(7 × 12), 3(14 × 15)
		3	2	95 × 69,85 × 97	8		2(20 × 14), 14 × 18, 3(8 × 17), 2(14 × 15)
		4	2	30 × 84,65 × 56	6		3(5 × 20), 2(12 × 13), 14 × 9
		5	3	75 × 63,42 × 55,73 × 89	5		5 × 9, 2(17 × 15), 2(11 × 9)
		6	3	90 × 57,67 × 52,76 × 86	10		3(20 × 15), 13 × 19, 3(16 × 5), 3(19 × 5)
		7	2	46 × 91,88 × 56	10		2(10 × 18), 14 × 9, 3(11 × 17), 3(17 × 9), 9 × 8
13	8	0	2	58 × 43,39 × 51	5	4	10 × 18, 3(9 × 9), 12 × 8
		1	3	94 × 47,97 × 39,85 × 70	6		8 × 8, 3(17 × 6), 2(15 × 6)
		2	2	84 × 72,85 × 77	6		13 × 18, 3(17 × 6), 2(5 × 13)
		3	3	83 × 81,55 × 67,81 × 86	7		12 × 12, 3(13 × 5), 15 × 11, 2(5 × 9)
		4	3	51 × 61,97 × 53,41 × 46	2		18 × 14, 6 × 8

(continued on next page)

Table A.13 (*continued*).

Inst.	P	s	Objects $\mathcal{O}_{sj}$		Items $\mathcal{I}_{si}$		
			m_s	$W_{sj} \times H_{sj}$	n_s	d	$w_{si} \times h_{si}$
		5	2	62 × 45,60 × 75	3		2(6 × 19), 6 × 16
		6	3	44 × 91,70 × 99,30 × 51	3		10 × 9, 2(11 × 7)
		7	3	96 × 85,41 × 59,98 × 73	5		3(18 × 9), 2(20 × 8)
		0	3	33 × 32,57 × 91,62 × 84	4		3(12 × 13), 10 × 10
		1	2	91 × 83,81 × 68	4		2(16 × 18), 16 × 7,15 × 8
		2	2	70 × 35,39 × 72	7		8 × 19, 2(10 × 10), 3(6 × 16), 10 × 6
14	8	3	2	78 × 92,51 × 93	5	1	20 × 14, 3(15 × 8), 16 × 17
		4	3	50 × 70,71 × 81,33 × 47	10		2(18 × 5), 13 × 15, 2(15 × 5), 3(17 × 5), 2(15 × 17)
		5	3	57 × 50,34 × 86,94 × 45	8		3(19 × 16), 3(18 × 12), 14 × 14,14 × 17
		6	3	68 × 94,50 × 68,48 × 53	11		3(5 × 5), 3(8 × 16), 3(14 × 12), 16 × 20,11 × 6
		7	2	61 × 64,73 × 89	6		3(7 × 6), 2(12 × 15), 16 × 5
		0	2	85 × 40,55 × 36	5		3(17 × 13), 2(8 × 11)
		1	3	59 × 53,92 × 88,51 × 58	10		2(18 × 12), 3(7 × 18), 3(11 × 17), 13 × 10, 8 × 11
		2	3	98 × 82, 2(44 × 49)	6		16 × 6, 3(18 × 17), 19 × 19,19 × 16
15	8	3	2	51 × 89,32 × 70	4	4	3(10 × 17), 13 × 20
		4	4	35 × 51,38 × 80, 2(31 × 49)	7		3(18 × 14), 2(15 × 8), 2(13 × 5)
		5	3	67 × 77,37 × 55,39 × 78	8		2(17 × 6), 3(10 × 6), 3(16 × 17)
		6	2	88 × 70,54 × 83	11		8 × 20, 2(11 × 11), 3(11 × 16), 2(15 × 10), 3(20 × 17)
		7	2	57 × 83,45 × 66	6		14 × 14, 3(16 × 10), 14 × 20,10 × 7
		0	5	31 × 98, 2(51 × 39), 2(30 × 64)	5		2(20 × 20), 2(20 × 17), 18 × 14
		1	2	86 × 87,82 × 98	4		3(7 × 9), 13 × 5
		2	3	68 × 97,65 × 65,78 × 34	10		2(17 × 13), 3(16 × 12), 12 × 11,6 × 17, 3(7 × 5)
16	8	3	2	54 × 85,53 × 59	4	2	12 × 6, 3(7 × 11)
		4	2	43 × 64,35 × 85	9		3(14 × 9), 3(16 × 17), 3(15 × 18)
		5	3	82 × 99,38 × 98,52 × 53	13		3(7 × 5), 3(9 × 10), 3(15 × 7), 13 × 10, 3(6 × 6)
		6	4	66 × 47, 3(35 × 41)	6		20 × 7, 2(19 × 12), 3(20 × 18)
		7	2	73 × 50,38 × 84	3		14 × 19, 2(17 × 11)
		0	2	81 × 37,33 × 64	6		2(9 × 15), 19 × 18, 3(11 × 14)
		1	3	34 × 83,59 × 86,72 × 44	5		20 × 15,14 × 10, 3(18 × 14)
		2	2	55 × 91,32 × 43	8		17 × 7, 2(14 × 20), 2(8 × 7), 3(8 × 18)
17	8	3	2	41 × 96,41 × 86	7	2	2(9 × 9), 18 × 7,15 × 16,17 × 18, 2(8 × 15)
		4	2	80 × 86,74 × 59	11		3(14 × 14), 3(6 × 20), 3(19 × 8), 2(11 × 12)
		5	4	85 × 39,85 × 63, 2(51 × 35)	10		2(20 × 16), 3(14 × 10), 2(18 × 20), 3(8 × 17)
		6	2	78 × 53,62 × 93	9		3(20 × 16), 2(11 × 5), 2(15 × 12), 14 × 14,9 × 14
		7	2	56 × 66,52 × 85	15		3(6 × 8), 3(8 × 5), 3(11 × 17), 3(12 × 16), 3(20 × 6)
		0	2	45 × 83,97 × 52	7		15 × 15, 3(11 × 13), 3(18 × 13)
		1	2	89 × 87,88 × 45	8		2(18 × 9), 6 × 7, 2(12 × 8), 8 × 19, 2(18 × 6)
		2	3	2(65 × 33), 92 × 72	8		3(19 × 20), 2(15 × 14), 3(9 × 14)
18	8	3	3	76 × 40,54 × 71,43 × 78	9	2	3(7 × 8), 5 × 17, 3(6 × 11), 2(17 × 15)
		4	2	72 × 74,89 × 73	5		3(11 × 7), 2(20 × 16)
		5	4	59 × 38, 2(44 × 32), 46 × 47	7		6 × 17, 2(18 × 16), 2(8 × 15), 2(18 × 11)
		6	3	56 × 41,100 × 45,40 × 92	2		13 × 20,18 × 13
		7	2	73 × 77,83 × 54	6		2(5 × 7), 2(16 × 18), 2(10 × 9)
		0	2	78 × 86,72 × 67	10		3(15 × 5), 3(6 × 6), 18 × 10, 2(8 × 10), 14 × 19
		1	3	53 × 67,37 × 80,67 × 56	8		2(17 × 5), 2(20 × 15), 2(15 × 13), 2(15 × 9)
		2	3	57 × 85,52 × 50,75 × 37	6		2(17 × 9), 2(9 × 9), 2(12 × 14)
19	8	3	3	64 × 44,45 × 96,75 × 52	10	2	3(18 × 20), 2(13 × 9), 8 × 9,9 × 7, 3(14 × 14)
		4	2	56 × 93,53 × 49	9		3(16 × 10), 3(10 × 14), 12 × 17, 2(6 × 15)
		5	2	51 × 89,65 × 72	5		16 × 14,18 × 8, 3(16 × 5)
		6	2	92 × 64,81 × 95	6		3(19 × 7), 2(6 × 14), 17 × 16
		7	3	62 × 52,32 × 97,95 × 35	8		3(7 × 16), 2(10 × 14), 11 × 12, 2(13 × 8)
		0	3	75 × 82,69 × 79,76 × 64	5		2(14 × 10), 2(15 × 13), 14 × 12
		1	2	49 × 68,61 × 79	12		3(11 × 18), 2(6 × 12), 2(7 × 7), 3(5 × 12), 2(13 × 18)
		2	3	92 × 41,74 × 51,78 × 93	7		10 × 5, 2(13 × 6), 2(8 × 10), 2(5 × 13)
20	8	3	3	61 × 85,45 × 51,34 × 50	7	3	3(8 × 19), 14 × 10, 3(9 × 11)
		4	2	41 × 50,63 × 84	6		2(13 × 20), 2(18 × 12), 2(10 × 5)
		5	2	81 × 43,53 × 45	6		2(7 × 14), 13 × 7, 2(9 × 11), 19 × 17
		6	3	35 × 82, 2(33 × 34)	7		6 × 14, 2(17 × 19), 19 × 10, 2(15 × 9), 11 × 11
		7	3	92 × 52,83 × 65,70 × 70	13		3(13 × 18), 2(16 × 6), 3(12 × 8), 3(5 × 18), 2(19 × 11)
21	12	0	2	65 × 50,93 × 92	5	2	2(7 × 8), 3(12 × 10)
		1	2	90 × 68,57 × 69	7		2(13 × 6), 3(19 × 14), 6 × 11, 6 × 5
		2	3	78 × 71,56 × 70,62 × 100	6		19 × 15, 2(8 × 17), 3(15 × 19)
		3	2	50 × 84,30 × 49	7		2(7 × 7), 14 × 17, 3(14 × 13), 8 × 16

(*continued on next page*)

Table A.13 (*continued*).

Inst.	P	s	Objects $\mathcal{O}_{sj}$		Items $\mathcal{I}_{si}$		
			m_s	$W_{sj} \times H_{sj}$	n_s	d	$w_{si} \times h_{si}$
		4	2	73 × 99,44 × 72	4		7 × 13, 3(8 × 7)
		5	3	48 × 50,70 × 79,100 × 52	10		17 × 16, 2(13 × 17), 2(5 × 10), 2(16 × 12), 3(6 × 15)
		6	3	36 × 93,36 × 77,92 × 90	4		2(13 × 15), 2(9 × 18)
		7	3	74 × 65,47 × 70,100 × 34	4		3(15 × 18), 11 × 9
		8	2	50 × 81,70 × 87	5		16 × 10, 2(16 × 17), 2(10 × 13)
		9	2	52 × 86,46 × 48	9		11 × 14, 2(19 × 8), 7 × 14, 2(15 × 6), 3(15 × 19)
		10	2	93 × 47,31 × 89	5		13 × 16,15 × 18, 3(18 × 7)
		11	2	81 × 92,37 × 80	11		3(9 × 14), 2(16 × 8), 2(5 × 19), 15 × 7, 3(14 × 17)
		0	2	73 × 35,72 × 91	5		6 × 5, 7 × 8,16 × 12, 2(11 × 8)
		1	2	39 × 63,54 × 63	8		2(13 × 13), 3(7 × 19), 3(11 × 7)
		2	3	96 × 44,63 × 56,54 × 53	5		8 × 20,15 × 11,18 × 8, 2(14 × 9)
		3	2	45 × 82,69 × 37	12		3(17 × 17), 3(19 × 11), 13 × 11, 3(9 × 11), 2(7 × 14)
		4	3	72 × 62,63 × 36,37 × 97	5		18 × 13,19 × 15, 2(18 × 19), 15 × 14
22	12	5	2	39 × 37,84 × 42	6	2	3(17 × 6), 3(10 × 5)
		6	3	2(31 × 38), 98 × 38	13		2(8 × 18), 3(8 × 16), 3(6 × 13), 2(16 × 7), 3(8 × 7)
		7	3	99 × 67,94 × 93,65 × 87	12		3(14 × 6), 20 × 19, 2(20 × 14), 3(17 × 17), 3(12 × 14)
		8	2	78 × 66,42 × 95	6		3(9 × 5), 18 × 13, 2(6 × 5)
		9	2	78 × 50,84 × 44	6		3(13 × 13), 12 × 9, 2(15 × 16)
		10	3	76 × 51,70 × 88,76 × 57	6		3(15 × 12), 3(7 × 12)
		11	3	71 × 40,44 × 52,55 × 58	6		5 × 18, 3(12 × 6), 2(6 × 17)
		0	3	100 × 62,68 × 83,86 × 66	4		3(5 × 11), 20 × 15
		1	2	82 × 51,65 × 68	8		2(8 × 19), 2(20 × 18), 19 × 11,14 × 7, 2(19 × 5)
		2	2	66 × 60,60 × 63	4		12 × 5, 3(17 × 14)
		3	3	81 × 52,32 × 97,97 × 46	7		2(20 × 10), 3(11 × 10), 2(13 × 18)
		4	2	34 × 57,39 × 95	6		2(13 × 18), 2(13 × 15), 6 × 12,20 × 17
23	12	5	2	38 × 92,33 × 95	6	2	2(19 × 9), 11 × 17, 2(17 × 9), 17 × 17
		6	3	77 × 44,37 × 100,50 × 37	9		3(9 × 16), 3(5 × 20), 3(19 × 9)
		7	3	86 × 62,92 × 99,72 × 43	5		2(19 × 5), 3(15 × 17)
		8	2	58 × 34,57 × 88	7		10 × 17,6 × 15, 2(5 × 12), 3(10 × 10)
		9	3	2(51 × 45), 50 × 53	9		3(19 × 6), 9 × 16, 5 × 8, 3(20 × 20), 15 × 10
		10	3	98 × 92,84 × 46,35 × 45	6		11 × 20, 2(12 × 15), 3(15 × 6)
		11	2	37 × 35,41 × 54	2		14 × 6,14 × 9
		0	3	69 × 73,63 × 95,62 × 94	8		19 × 20, 3(13 × 12), 14 × 7, 3(14 × 19)
		1	2	69 × 32,39 × 59	8		8 × 9, 2(10 × 8), 3(18 × 14), 2(10 × 19)
		2	3	97 × 33,78 × 42,56 × 30	7		17 × 14, 3(15 × 10), 3(20 × 12)
		3	3	87 × 55,36 × 76,33 × 56	4		3(10 × 6), 15 × 20
		4	3	100 × 84, 2(36 × 41)	10		15 × 18, 3(8 × 8), 2(13 × 16), 20 × 15, 3(15 × 17)
24	12	5	3	85 × 67,92 × 35,46 × 98	5	2	8 × 19,19 × 6, 3(19 × 19)
		6	2	52 × 75,56 × 60	10		3(14 × 18), 3(8 × 6), 5 × 15, 3(9 × 17)
		7	3	35 × 53,67 × 54,62 × 93	4		11 × 7, 3(9 × 7)
		8	2	97 × 66,69 × 39	4		7 × 18,8 × 8, 2(19 × 17)
		9	2	83 × 38,54 × 66	7		2(18 × 7), 3(20 × 13), 2(19 × 17)
		10	2	87 × 51,33 × 55	4		2(9 × 20), 2(15 × 7)
		11	3	68 × 68,39 × 87,82 × 78	6		19 × 14, 2(5 × 18), 3(13 × 8)
		0	3	86 × 45,57 × 40,64 × 87	9		15 × 11, 3(14 × 20), 3(9 × 16), 2(15 × 7)
		1	2	70 × 31,95 × 99	8		7 × 6, 2(12 × 20), 19 × 8, 3(15 × 8), 7 × 18
		2	3	49 × 36,83 × 98,35 × 51	4		2(10 × 16), 2(20 × 12)
		3	4	61 × 63,97 × 89, 2(34 × 40)	12		20 × 15, 3(14 × 18), 3(16 × 15), 3(9 × 6), 2(8 × 16)
		4	3	33 × 65,68 × 56,90 × 82	10		3(12 × 11), 3(20 × 13), 12 × 20, 3(6 × 13)
25	12	5	2	83 × 83,79 × 81	5	1	3(15 × 19), 11 × 14,11 × 15
		6	2	51 × 77,33 × 95	6		2(5 × 5), 2(7 × 12), 2(8 × 14)
		7	2	32 × 35,99 × 81	6		2(17 × 17), 3(14 × 7), 7 × 13
		8	3	47 × 58,72 × 81,83 × 51	2		14 × 6,5 × 17
		9	3	42 × 99,75 × 47,57 × 87	10		2(6 × 20), 2(15 × 6), 3(17 × 14), 19 × 14, 2(19 × 12)
		10	2	66 × 59,54 × 86	4		5 × 18, 3(5 × 20)
		11	3	55 × 58,99 × 45,67 × 73	6		2(11 × 15), 3(20 × 13), 13 × 19
		0	2	51 × 42,79 × 85	5		6 × 13,8 × 15, 2(16 × 7), 15 × 15
		1	3	95 × 82,100 × 90,54 × 75	3		2(18 × 5), 7 × 17
		2	2	85 × 35,69 × 83	4		7 × 19, 3(17 × 13)
		3	2	90 × 100,81 × 96	11		2(13 × 12), 2(12 × 19), 2(20 × 17), 2(16 × 19), 3(14 × 6)
		4	3	79 × 91,51 × 40,85 × 79	8		13 × 15,19 × 7, 2(14 × 15), 2(6 × 19), 2(20 × 7)
26	12	5	3	78 × 59,85 × 31,85 × 56	10	5	2(17 × 11), 3(10 × 9), 5 × 19, 3(15 × 11), 18 × 12
		6	2	81 × 76,66 × 70	5		2(12 × 6), 2(19 × 16), 11 × 20

(*continued on next page*)

Table A.13 (*continued*).

Inst.	P	s	Objects $\mathcal{O}_{sj}$		Items $\mathcal{I}_{si}$		
			m_s	$W_{sj} \times H_{sj}$	n_s	d	$w_{si} \times h_{si}$
		7	2	80 × 52,74 × 68	3		14 × 6,14 × 17,13 × 14
		8	3	83 × 95,45 × 48,95 × 63	5		7 × 10, 3(19 × 8), 18 × 16
		9	2	79 × 82,79 × 36	7		2(17 × 19), 2(13 × 11), 3(6 × 10)
		10	3	32 × 85,45 × 97,78 × 86	8		2(14 × 18), 3(17 × 19), 2(12 × 15), 7 × 13
		11	2	45 × 42,36 × 71	7		9 × 15, 3(14 × 8), 3(19 × 10)
27	12	0	5	47 × 71,71 × 96, 3(32 × 51)	10	1	16 × 9, 3(19 × 13), 3(17 × 12), 3(18 × 17)
		1	2	62 × 65,38 × 91	3		2(20 × 18), 11 × 5
		2	2	100 × 62,69 × 62	7		18 × 5, 3(13 × 19), 3(17 × 15)
		3	2	61 × 47,84 × 91	11		6 × 6, 3(20 × 5), 15 × 12, 3(17 × 18), 3(7 × 15)
		4	3	90 × 82,42 × 52,91 × 35	12		3(13 × 13), 5 × 18, 3(8 × 8), 2(9 × 15), 3(10 × 18)
		5	2	93 × 96,95 × 54	11		2(8 × 15), 2(16 × 15), 15 × 13, 3(11 × 5), 3(10 × 5)
		6	2	67 × 97,72 × 65	5		3(9 × 18), 2(14 × 14)
		7	2	43 × 81,58 × 100	5		2(11 × 6), 18 × 17,9 × 7,8 × 13
		8	3	37 × 58,48 × 40,54 × 93	4		16 × 20, 3(10 × 13)
		9	3	63 × 69,71 × 52,50 × 36	4		2(15 × 17), 2(19 × 19)
		10	2	89 × 50,94 × 56	8		3(5 × 5), 14 × 11,13 × 11, 3(5 × 20)
		11	2	91 × 67,57 × 72	7		15 × 17,18 × 16, 2(7 × 18), 3(13 × 19)
28	12	0	2	93 × 73,38 × 66	9	3	3(15 × 13), 13 × 11, 3(15 × 5), 2(8 × 15)
		1	3	94 × 36,53 × 41,100 × 64	5		2(20 × 16), 3(6 × 12)
		2	2	69 × 98,92 × 99	8		2(17 × 19), 3(8 × 10), 3(8 × 17)
		3	3	75 × 42,36 × 41,66 × 47	3		2(19 × 12), 14 × 17
		4	3	2(35 × 40), 59 × 64	9		19 × 11,17 × 11,6 × 20, 3(18 × 17), 3(11 × 6)
		5	2	71 × 51,53 × 31	6		3(19 × 14), 3(15 × 15)
		6	2	73 × 55,71 × 61	6		2(14 × 18), 2(5 × 19), 2(15 × 16)
		7	2	93 × 34,35 × 74	5		2(12 × 17), 9 × 15, 2(19 × 9)
		8	3	99 × 49, 2(37 × 69)	14		3(14 × 5), 2(7 × 5), 3(15 × 15), 3(19 × 18), 3(9 × 19)
		9	2	65 × 81,31 × 61	12		3(11 × 13), 3(7 × 8), 3(6 × 15), 3(6 × 9)
		10	2	79 × 48,75 × 73	4		20 × 19, 3(12 × 7)
		11	2	89 × 72,58 × 91	12		2(15 × 14), 2(10 × 17), 2(7 × 18), 3(11 × 20), 3(15 × 18)
29	12	0	3	70 × 66,90 × 86,36 × 44	7	1	12 × 20, 2(8 × 20), 15 × 16, 2(9 × 6), 12 × 9
		1	3	75 × 85,47 × 59,32 × 38	6		14 × 19,8 × 11,7 × 10, 3(6 × 5)
		2	3	99 × 44,45 × 83,65 × 95	5		10 × 6,15 × 20, 3(16 × 10)
		3	3	86 × 72,48 × 81,72 × 42	4		9 × 12,10 × 12,11 × 14,7 × 14
		4	2	99 × 35,48 × 43	6		5 × 5, 2(10 × 11), 3(6 × 10)
		5	3	39 × 43,72 × 55,52 × 60	6		2(18 × 12), 2(11 × 6), 5 × 15,9 × 13
		6	2	30 × 34,81 × 84	4		17 × 10, 3(6 × 7)
		7	3	81 × 48,46 × 32,38 × 36	9		9 × 15,11 × 9, 3(5 × 18), 2(13 × 12), 2(13 × 6)
		8	3	89 × 65,99 × 66,46 × 66	6		5 × 9, 2(8 × 16), 11 × 5,6 × 16,10 × 11
		9	3	40 × 92,46 × 49,70 × 67	8		19 × 15,20 × 15, 3(8 × 17), 3(12 × 10)
		10	3	76 × 42,66 × 90,85 × 60	10		2(9 × 9), 3(11 × 14), 3(20 × 9), 2(14 × 14)
		11	5	91 × 86, 2(46 × 39), 2(41 × 41)	11		3(16 × 20), 2(19 × 16), 3(6 × 7), 3(20 × 15)
30	12	0	3	34 × 50,34 × 38,98 × 33	3	2	2(6 × 7), 16 × 8
		1	3	49 × 78,53 × 70,84 × 100	2		8 × 19,9 × 14
		2	3	79 × 96,69 × 43,76 × 73	8		20 × 5, 3(5 × 7), 17 × 10, 2(12 × 12), 5 × 13
		3	2	50 × 98,60 × 59	9		2(5 × 8), 3(20 × 13), 2(18 × 16), 2(13 × 15)
		4	3	36 × 100,90 × 41,73 × 97	5		8 × 15,16 × 19, 2(17 × 11), 7 × 7
		5	3	82 × 96,51 × 40,55 × 47	6		3(9 × 8), 20 × 18, 2(10 × 9)
		6	3	50 × 78,77 × 35,66 × 79	4		3(9 × 7), 11 × 10
		7	2	44 × 45,76 × 54	11		8 × 17, 3(11 × 7), 3(8 × 20), 12 × 14, 3(14 × 11)
		8	3	62 × 71,93 × 67,90 × 93	4		15 × 13, 3(15 × 15)
		9	3	89 × 62,75 × 86,63 × 40	3		17 × 9, 2(8 × 18)
		10	3	38 × 59,59 × 71,100 × 51	4		15 × 13, 3(10 × 5)
		11	5	35 × 99, 2(46 × 94), 2(61 × 51)	10		3(19 × 16), 4(15 × 20), 3(18 × 17)

References

Ali, R., Muhammad, S., & Takahashi, R. H. C. (2021). Decision making viva genetic algorithm for the utilization of leftovers. *International Journal of Intelligent Systems*, *36*, 1746–1769. http://dx.doi.org/10.1002/int.22359.

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. http://dx.doi.org/10.1111/itor.12077.

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. http://dx.doi.org/10.1057/jors.2013.108.

Ayadia, O., Masmoudib, M., Ameura, M. B., & Masmoudi, F. (2017). A new PSO-based algorithm for two-dimensional non-guillotine non-oriented cutting stock problem. *Applied Artificial Intelligence*, *31*, 376–393. http://dx.doi.org/10.1080/08839514.2017.1346966.

Baykasoglu, A., & Özbel, B. K. (2021). Modeling and solving a real-world cutting stock problem in the marble industry via mathematical programming and stochastic diffusion search approaches. *Computers & Operations Research*, *128*, Article 105173. http://dx.doi.org/10.1016/j.cor.2020.105173.

Birgin, E. G., Romão, O. C., & Ronconi, D. P. (2020). The multiperiod two-dimensional non-guillotine cutting stock problem with usable leftovers. *International Transactions in Operational Research*, *27*, 1392–1418. http://dx.doi.org/10.1111/itor.12648.

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 K. Chan, & J. Yeh (Eds.), *Advances* *in engineering research*: *vol. 13*, *Proceedings of the 2015 international conference on electrical, automation and mechanical engineering* (pp. 776–779). Atlantis Press, http://dx.doi.org/10.2991/eame-15.2015.208.

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. http://dx.doi.org/10.1111/j.1475-3995.2012.00868.x.

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. http://dx.doi.org/10.1016/j.ejor.2013.11.026.

do Nascimento, D. N., de Araujo, S. A., & Cherri, A. C. (2022). Integrated lot-sizing and one-dimensional cutting stock problem with usable leftovers. *Annals of Operations Research*, *316*, 785–803. http://dx.doi.org/10.1007/s10479-020-03772-9.

Dyckhoff, H. (1981). A new linear programming approach to the cutting stock problem. *Operations Research*, *29*, 1092–1104. http://dx.doi.org/10.1287/opre.29.6.1092.

Poldi, K. C., & Arenales, M. N. (2010). O problema de corte de estoque unidimensional multiperíodo. *Pesquisa Operacional*, *30*, 153–174. http://dx.doi.org/10.1590/S0101-74382010000100008.

Powell, W. B. (2007). *Approximate dynamic programming: Solving the courses of dimensionality*. Hoboken, NJ: Wiley.

Ronconi, D. P., & Powell, W. B. (2010). Minimizing total tardiness in a stochastic single machine scheduling problem using approximate dynamic programming. *Journal of Scheduling*, *13*, 597–607. http://dx.doi.org/10.1007/s10951-009-0160-6.

Roodman, G. M. (1986). Near-optimal solutions to one-dimensional cutting stock problems. *Computers & Operations Research*, *13*, 713–719. http://dx.doi.org/10.1016/0305-0548(86)90077-8.

Scheithauer, G. (1991). A note on handling residual lengths. *Optimization*, *22*, 461–466. http://dx.doi.org/10.1080/02331939108843687.

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. http://dx.doi.org/10.1016/j.ejor.2010.01.039.

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. http://dx.doi.org/10.1057/jors.2013.25.

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. http://dx.doi.org/10.1007/s10100-017-0466-y.

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. http://dx.doi.org/10.1504/EJIE.2016.078141.