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

Ernesto G. Birgin[†] Oberlan C. Romão[‡] Débora P. Ronconi[§]

January 28, 2022

Abstract

In [E. G. Birgin, O. C. Romão, and D. P. Ronconi, The multi-period two-dimensional non-guillotine cutting stock problem with usable leftovers, *International Transactions in Operational Research* 27(3), 1392–1418, 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 approach's performance is compared to the performance of a myopic technique. Numerical experiments show the efficacy of the proposed approach.

Key words: Two-dimensional cutting stock with usable leftovers, non-guillotine cutting and packing, multi-period scenario, forward-looking or looking-ahead approach, matheuristic.

1 Introduction

In this paper, we consider the multi-period two-dimensional non-guillotined 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 $t_0 < t_1 < \dots < t_P$ are

*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, 306083/2016-7 and 302682/2019-8).

[†]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. e-mail: egbirgin@ime.usp.br. Corresponding author.

[‡]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. e-mail: oberlan.romao@ufes.br

[§]Department of Production Engineering, Polytechnic School, University of São Paulo, Av. Luciano Gualberto, 1380, Cidade Universitária, 05508-010, São Paulo SP, Brazil. e-mail: dronconi@usp.br

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 $t_{s-1} \leq t \leq t_s$ 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 can not 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. 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.

In the current work, we propose a forward-looking matheuristic to solve medium- and large-sized instances of the problem described in the paragraph above. 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 required to produced the period ordered items. The estimation of 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. The larger the durability of the leftovers, the greater the opportunity for economy. 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 Cherri et al. (2013, 2014); Poldi and Arenales (2010); Tomat and Gradišar (2017); Baykasoglu and Özbel (2021); Ali et al. (2021); do Nascimento et al. (2021). 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., 2021) 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 and 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.

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 Figure 1. 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 $s = p, \dots, P-1$, 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 ($j = 1, \dots, m_s$) and a set of n_s ordered items $\mathcal{I}_{si}$ with width w_{si} and height h_{si} ($i = 1, \dots, n_s$). A catalogue composed by d items $\bar{\mathcal{I}}_i$ with width $\bar{w}_i$ and height $\bar{h}_i$ ($i = 1, \dots, d$) 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

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

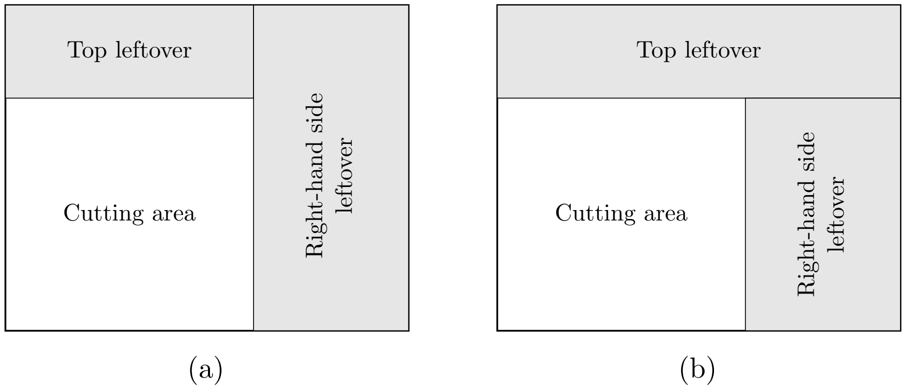


Figure 1: Pictures (a) and (b) illustrate the two possible ways in which two leftovers can be generated from an object by performing a vertical and a horizontal guillotine pre-cut. In case (a), the vertical guillotine pre-cut is made first; while, in case (b), the horizontal guillotine pre-cut is made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

where

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

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 $p, \dots, P-1$ making use of leftovers; and, among all solutions with minimum cost, maximizing the value of the usable leftovers at instant P. See Figures 2 and 3. Figure 2 describes a toy instance of the problem; while Figure 3 exhibits two different feasible solutions.

Purchasable objects $\mathcal{O}_{sj}$ ($s = p, \dots, P-1$, $j = 1, \dots, m_s$) 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 to each (purchasable or leftover) object $\mathcal{O}_{sj}$ ($s = p, \dots, P$, $j = 1, \dots, \bar{m}_s$) 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 $j_1^s \leq j_2^s \leq \dots \leq j_{\hat{m}_s}^s$ 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 $c_{P,j}$ ($j = m_P + 1, \dots, \bar{m}_P$) 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 ($s = p, \dots, P-1$) 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 $j_1^s, j_2^s, \dots$ 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

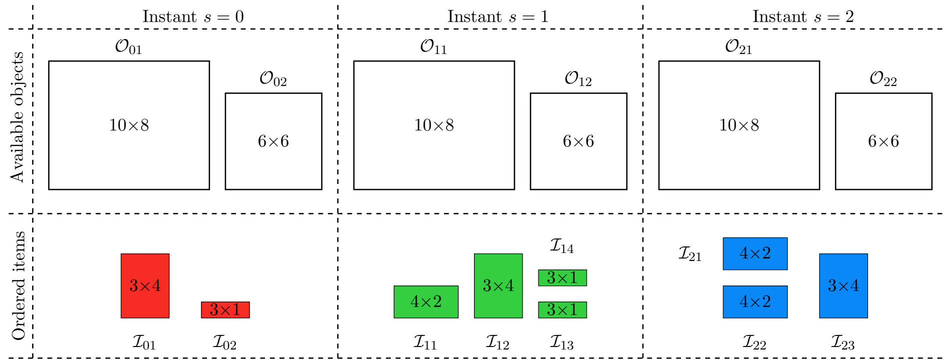


Figure 2: Illustration of a small instance with $p = 0$, $P = 3$, and $\xi = P - p = 3$, meaning that usable leftovers generated at any period remain usable up to instant P. The picture shows the available purchasable objects and the ordered items at each instant $s \in \{0, 1, 2\}$. The numbers of available 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 $\bar{h}_1 = 1$.

setting $\hat{m}_s = k$.

The description of the problem's variables follows. Variables $v_{sij} \in \{0, 1\}$ ($s = p, \dots, P - 1$, $j = 1, \dots, \bar{m}_s$, $i = 1, \dots, n_s$) 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 $u_{sj} \in \{0, 1\}$ ($s = p, \dots, P - 1, j = 1, \dots, m_s$) identify whether at least an item is assigned to object $\mathcal{O}_{sj}$ or not ($u_{sj} = 1$ and $u_{sj} = 0$, respectively). Variables $\eta_{sj} \in \{0, 1\}$ ($s = p, \dots, P - 1, j = 1, \dots, \bar{m}_s$) 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 $r_{sj} \in \mathbb{R}$ ($s = p, \dots, P - 1$, $j = 1, \dots, \bar{m}_s$) 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 $\bar{H}_{sj} \in \mathbb{R}$ ($s = p, \dots, P$, $j = 1, \dots, \bar{m}_s$) 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 $\tau_{sii'} \in \{0, 1\}$ ($s = p, \dots, P - 1$, $i = 1, \dots, n_s$, $i' = i + 1, \dots, n_s$) are auxiliary variables used to avoid the overlapping between items. Variables $\gamma_j \in \mathbb{R}$ ($j = 1, \dots, \bar{m}_P$) 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 $\omega_{j\ell} \in \mathbb{R}$ ($j = 1, \dots, \bar{m}_P$, $\ell = 1, \dots, L$) are auxiliary variables used to linearize the computation of these areas (product of the leftovers variable dimensions), where $L = \lfloor \log_2(\hat{W}) \rfloor + 1$, $\hat{W} = \max\{W_{sj} \mid s = p, \dots, P - 1, j = 1, \dots, m_s\}$, and, for further reference, $\hat{H} = \max\{H_{sj} \mid s = p, \dots, P - 1, j = 1, \dots, m_s\}$. The auxiliary variables $\zeta_{ji} \in \{0, 1\}$ ($j = 1, \dots, \bar{m}_P$, $i = 1, \dots, d$) are used to nullify the value of the area of a leftover at instant P if it can not fit any item from the catalogue.

The problem consists in minimizing

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

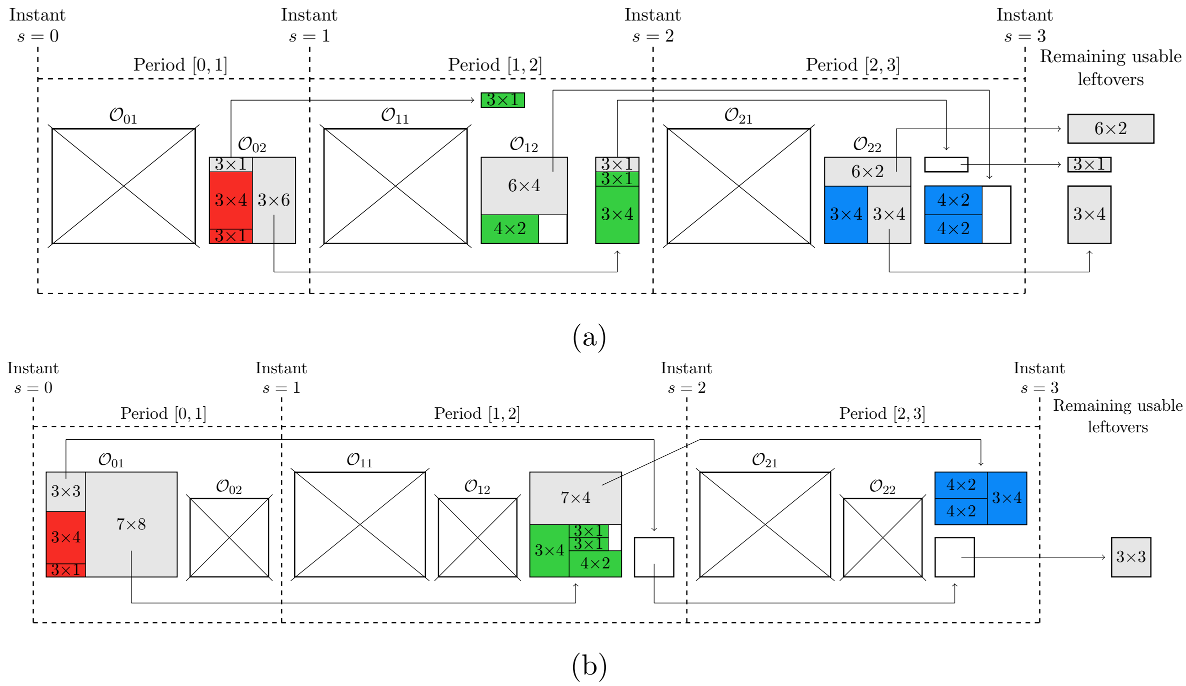


Figure 3: Illustration of two solutions that, at each period, may cut ordered items from purchasable objects or from usable leftovers from previous periods. (a) Greedy solution obtained by a myopic method that, *at each decision instant*, minimizes the cost of the purchasable objects required to cut 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).

subject to

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

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

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

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

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

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

$$\begin{array}{rcccl}
0 & \le & \bar{H}_{s+1,\ell_1} & \le & \hat{H}u_{sj}, \\
t_{sj} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_1} & \le & t_{sj} + (1-u_{sj})\hat{H}, \\[4pt]
0 & \le & \bar{W}_{s+1,\ell_1} & \le & \hat{W}u_{sj}, \\
\bar{W}_{sj} - r_{sj} - (1-\eta_{sj})\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} - r_{sj} + (1-\eta_{sj})\hat{W} + (1-u_{sj})\hat{W}, \\
\bar{W}_{sj} - \eta_{sj}\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} + \eta_{sj}\hat{W} + (1-u_{sj})\hat{W}, \\[4pt]
0 & \le & \bar{W}_{s+1,\ell_2} & \le & \hat{W}u_{sj}, \\
r_{sj} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_2} & \le & r_{sj} + (1-u_{sj})\hat{W}, \\[4pt]
0 & \le & \bar{H}_{s+1,\ell_2} & \le & \hat{H}u_{sj}, \\
\bar{H}_{sj} - (1-\eta_{sj})\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} + (1-\eta_{sj})\hat{H} + (1-u_{sj})\hat{H}, \\
\bar{H}_{sj} - t_{sj} - \eta_{sj}\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} - t_{sj} + \eta_{sj}\hat{H} + (1-u_{sj})\hat{H},
\end{array} \tag{10}$$

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

$$\begin{array}{rcccl}
\bar{H}_{sj} - \hat{H}u_{sj} & \le & \bar{H}_{s+1,\ell_1} & \le & \bar{H}_{sj} + \hat{H}u_{sj}, \\
t_{sj} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_1} & \le & t_{sj} + (1-u_{sj})\hat{H}, \\[4pt]
\bar{W}_{sj} - \hat{W}u_{sj} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} + \hat{W}u_{sj}, \\
\bar{W}_{sj} - r_{sj} - (1-\eta_{sj})\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} - r_{sj} + (1-\eta_{sj})\hat{W} + (1-u_{sj})\hat{W}, \\
\bar{W}_{sj} - \eta_{sj}\hat{W} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_1} & \le & \bar{W}_{sj} + \eta_{sj}\hat{W} + (1-u_{sj})\hat{W}, \\[4pt]
0 & \le & \bar{W}_{s+1,\ell_2} & \le & \hat{W}u_{sj}, \\
r_{sj} - (1-u_{sj})\hat{W} & \le & \bar{W}_{s+1,\ell_2} & \le & r_{sj} + (1-u_{sj})\hat{W}, \\[4pt]
0 & \le & \bar{H}_{s+1,\ell_2} & \le & \hat{H}u_{sj}, \\
\bar{H}_{sj} - (1-\eta_{sj})\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} + (1-\eta_{sj})\hat{H} + (1-u_{sj})\hat{H}, \\
\bar{H}_{sj} - t_{sj} - \eta_{sj}\hat{H} - (1-u_{sj})\hat{H} & \le & \bar{H}_{s+1,\ell_2} & \le & \bar{H}_{sj} - t_{sj} + \eta_{sj}\hat{H} + (1-u_{sj})\hat{H},
\end{array} \tag{11}$$

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

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

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

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

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

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

and

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

The objective function (3) is given by the cost of the used *purchasable* objects multiplied by an 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 (13,14,15,16) model the value γ_j of the j-th 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 (13,14,15,16), 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 *area* of the leftovers of the last instant of the considered horizon plays 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 area.

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 $j=1,\dots,\bar{m}_\kappa$ 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 Figure 3(a) to the instance described in Figure 2. Its flaw is to ignore the effect in the future of the decisions made at each instant κ. Figure 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 Figure 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 catalog). 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) can not 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

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

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. Figure 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 Figure 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 can not 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 Figure 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 $\mathcal{O}_{02}$.

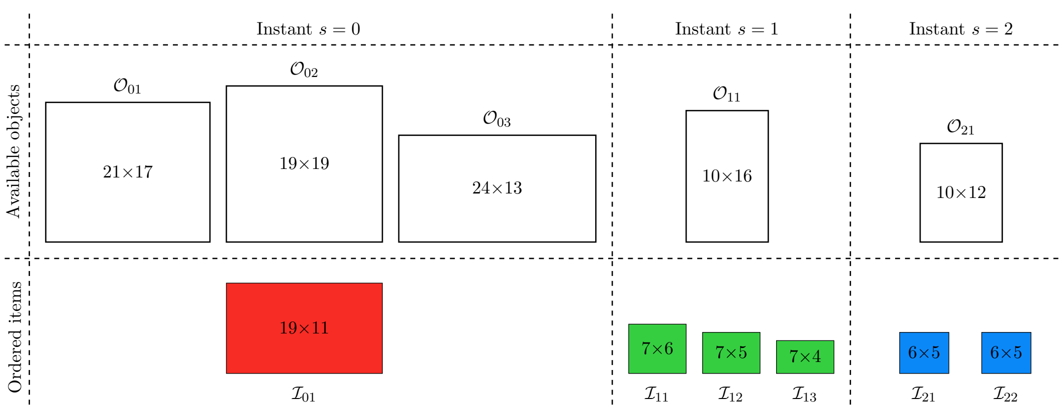
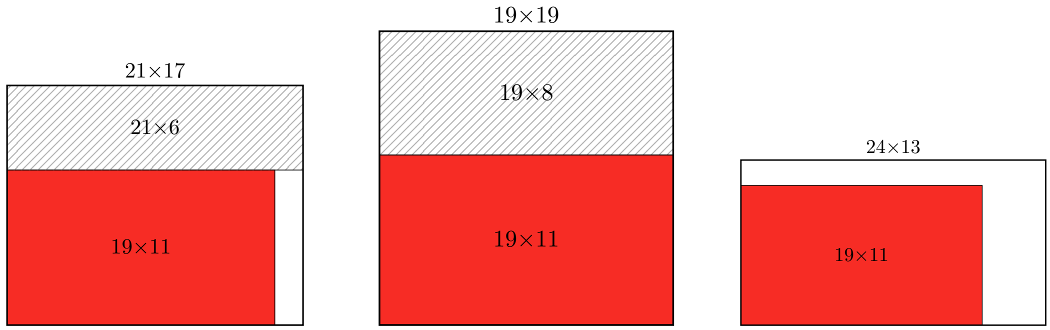


Figure 4: Illustration of a small instance with $p = 0$, $P = 3$. The figure displays the available purchasable objects and the ordered items at each instant $s \in \{p, \ldots, P-1\}$.

If the myopic approach is applied to the instance of Figure 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

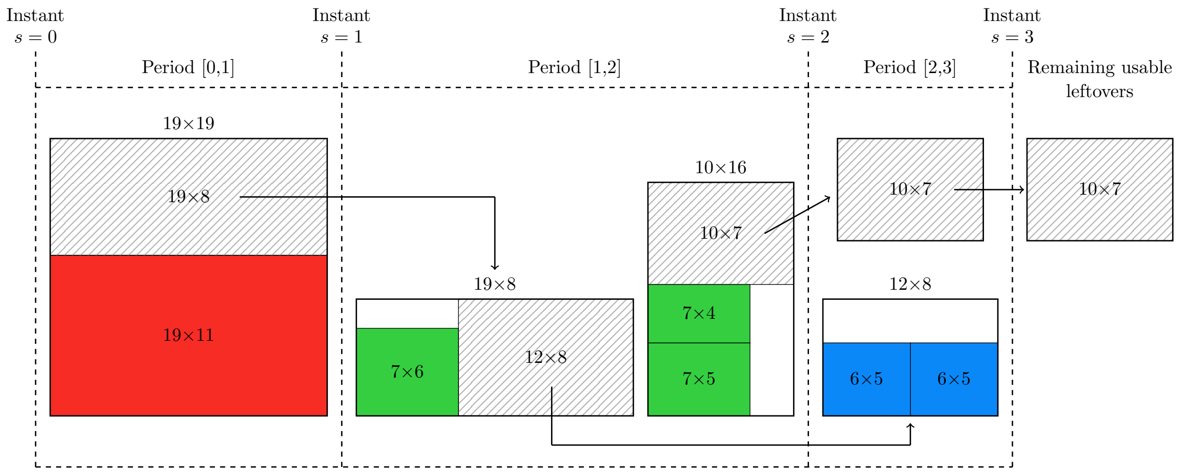


(a) $\mathcal{O}_{01}$ (b) $\mathcal{O}_{02}$ (c) $\mathcal{O}_{03}$

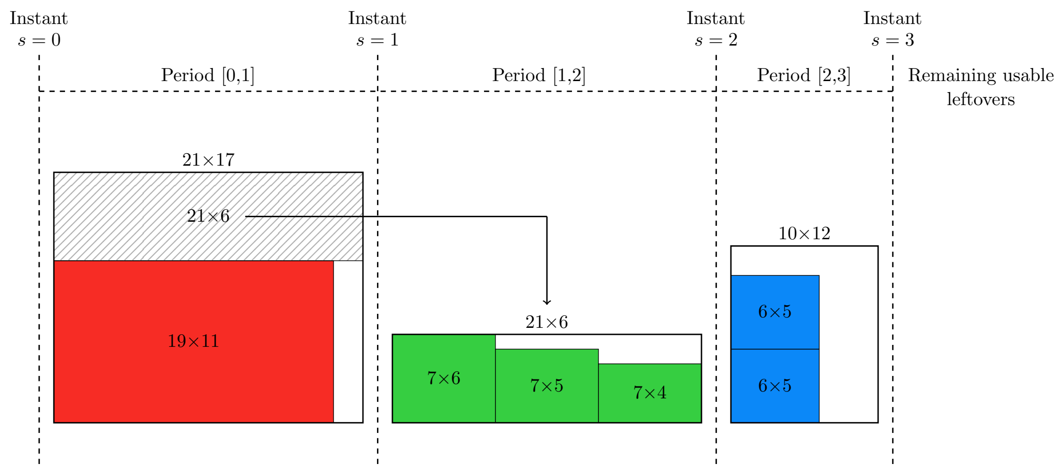
Figure 5: Dashed regions represent the usable leftovers in the assignment of item $\mathcal{I}_{01}$ to the three purchasable objects available at instant $s = 0$.

will be used in forthcoming periods, is used, then the solution found is the one illustrated in Figure 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 Figure 5(b) becomes $361 - 102 = 259$. The amortized cost of object $\mathcal{O}_{01}$ (for the configuration in Figure 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 Figure 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 Figure 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 Figure 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



(a)



(b)

Figure 6: Different feasible solutions to the instance of Figure 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.

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

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

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. Before solving the single-period problem $\mathcal{M}(\delta, \kappa, \kappa+1)$ we proceed as follows. (The supra-index η will be omitted for simplicity.) Let $j^\kappa_1 \le j^\kappa_2 \le \dots \le j^\kappa_{\hat{m}_\kappa}$ be the indices of the $\hat{m}_\kappa$ objects that generate leftovers, that correspond to the indices j of objects $\mathcal{O}_{\kappa j}$ ($j = 1, \dots, \bar{m}_\kappa$) such that $e_{\kappa j} > 0$. On the one hand, every $j_k \le m_\kappa$ is a purchasable object. This means that its two leftovers are first-order leftovers. So, in this case, we initialize the used area of the two leftovers as

$$a_{\kappa+1, m_{\kappa+1}+2k-1} = a_{\kappa+1, m_{\kappa+1}+2k} = 0$$

and the ancestor (or origin) of the two leftovers as themselves, i.e.

$$o_{\kappa+1, m_{\kappa+1}+2k-1} = m_{\kappa+1} + 2k - 1 \text{ and } o_{\kappa+1, m_{\kappa+1}+2k} = m_{\kappa+1} + 2k.$$

On the other hand, every $j_k > m_\kappa$ is a leftover that is generating high-order leftovers. So, in this case, we simply set the ancestor (or origin) of the two leftovers as

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

(Note that the "ancestor" is a pair that saves the instant and the index of the first-order leftover that generated the high-order leftover.) After these initializations, we are ready to solve the single-period problem $\mathcal{M}(\delta, \kappa, \kappa+1)$. After solving it, we can also set the area of the two first-order leftovers as

$$A_{\kappa+1, m_{\kappa+1}+2k-1} = \gamma_{m_{\kappa+1}+2k-1} \text{ and } A_{\kappa+1, m_{\kappa+1}+2k} = \gamma_{m_{\kappa+1}+2k},$$

for every $j_k \le m_\kappa$. Then, for each item $\mathcal{I}_{\kappa i}$ ($i = 1, \dots, n_\kappa$), 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 such that $v_{\kappa ij} = 1$. If $j > m_\kappa$, then 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 the leftover $\mathcal{O}_{\kappa j}$ (that may be itself or not), i.e.

$$a_{o_{\kappa j}} \leftarrow a_{o_{\kappa j}} + w_{\kappa i} \times h_{\kappa i}.$$

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

$$f^\eta_{\kappa+1,j} = \frac{a_{\kappa+1,j}}{A_{\kappa+1,j}} \text{ for } \kappa = p, \dots, P-1 \text{ and } j = m_{\kappa+1}, \dots, 2m_\kappa.$$

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.

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 enable. Numerical experiments were conducted using a machine with Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz with 160GB of RAM memory, and Ubuntu Server 18.04 operating system. Single-period instances within the myopic and the forward-looking approaches were solved using IBM ILOG CPLEX 12.10.0. A solution is reported as optimal by CPLEX when

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

or

$$\text{relative gap} = \frac{|\text{best feasible solution} - \text{best lower bound}|}{10^{-10} + |\text{best feasible solution}|} \leq \varepsilon_{\text{rel}}, \tag{27}$$

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 $\kappa = p, \dots, P-1$, 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.

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). 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 $\xi \in \{1, 2, 3, 4\}$. The experiments in Birgin et al. (2020) show that, when applied to these one hundred instances, CPLEX found an optimal solution in 91 cases. Therefore, we applied the forward-looking approach with all combinations of δ_{ini} and $\sigma \in \{0.5, 0.55, \dots, 1.0\}$ to these 91 instances and computed the gap to the known optimal solution computed by CPLEX.

Figure 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 a small variation in the average results of the method. It should be noted that the number of cycles (or iterations) η that are performed until the satisfaction of the stopping rule (26) depends on $\delta_{\rm ini}$ and σ. Figure 7 (middle and bottom) displays the average number of cycles η and the average elapsed CPU time in seconds, as a function of $\delta_{\rm ini}$ and σ. On the one hand, the CPU time has a low dependence on σ and, roughly speaking, is an increasing function of $\delta_{\rm ini}$. On the other hand, the number of cycles has a low dependence on $\delta_{\rm ini}$ 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. Figure 7 shows that, actually, this phenomenon occur; but it produces a premature stopping with lower quality solutions. However, regardless of the metrics related to computational cost, based on the quality of the solutions obtained, we selected $(\delta_{\rm ini}, \sigma) = (0.9, 0.9)$ for the rest of the experiments.

4.2 Forward-looking versus myopic approach

In a second set of experiments, we compare the introduced forward-looking approach with $(\delta_{\rm ini}, \sigma) = (0.9, 0.9)$ 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 is considered. Instances were generated with the random generator introduced in Birgin et al. (2020). In order to allow reproducibility, a table describing each instance is given in the Appendix. Table 1 shows the number of binary variables, continuous variables, and constraints of each instance when $\xi \in \{1, 2, 3, 4\}$ and, for the instances with eight or twelve periods, $\xi = P$. 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

$$100 \left(\frac{F_{\rm flook} - F_{\rm myopic}}{F_{\rm myopic}} \right) \%, \tag{28}$$

where $F_{\rm flook}$ is the best objective function value found by the forward-looking approach and $F_{\rm myopic}$ 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 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

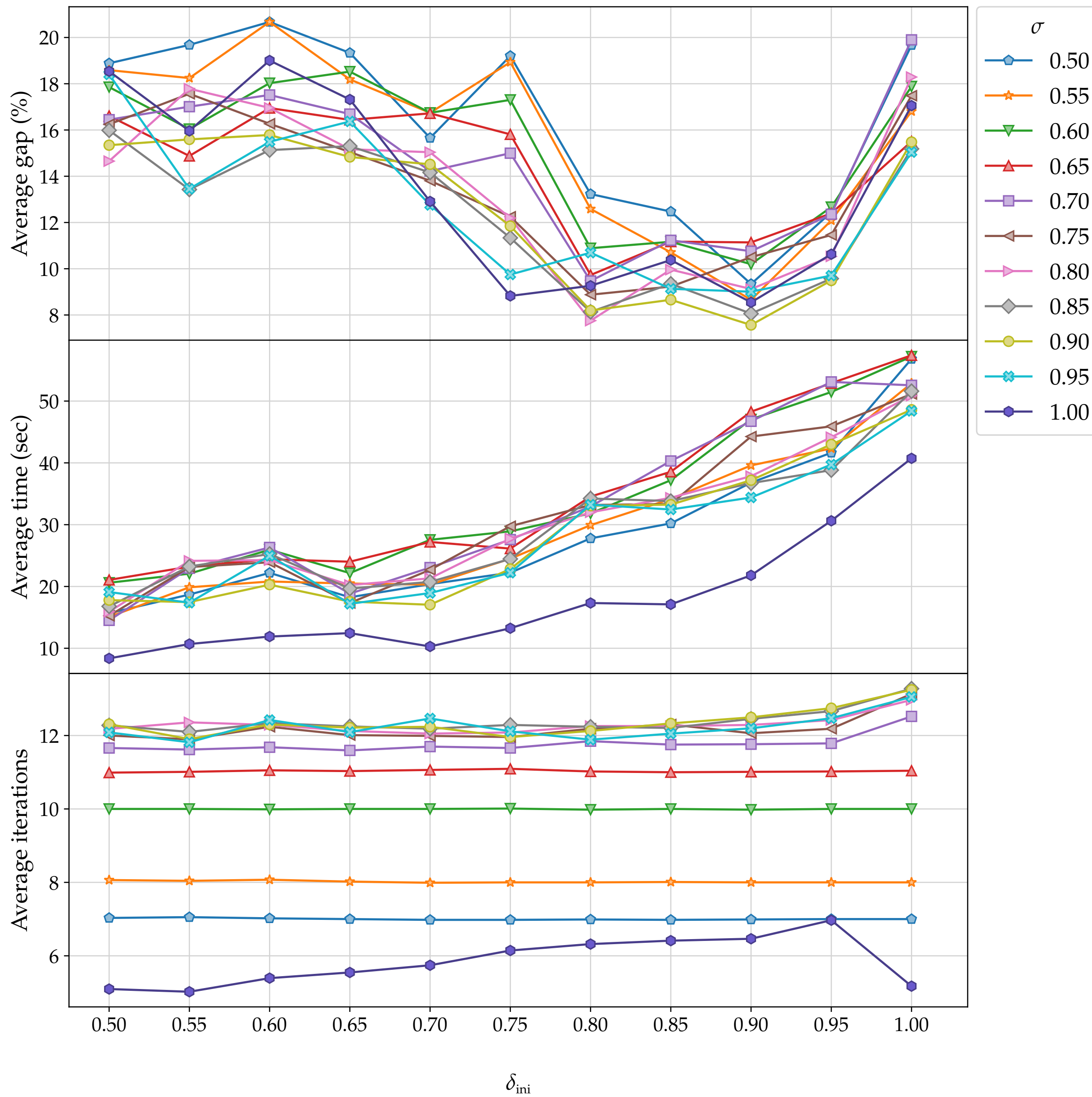


Figure 7: Average gap (to optimal solution computed with CPLEX), CPU time (in seconds), and number of cycles of the forward-looking approach for variations of its parameters δ_{ini} and σ.

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 "W/T/L G(%)", i.e. for each combination of number of periods $P \in \{4, 8, 12\}$ and

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	Since instances from 1 to 10 have $P = 4$ periods, the case $\xi = P$ coincides with the case $\xi = 4$.		
	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			
	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

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 (28). 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. 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, 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 economy using 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

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	60.1	400,703,843	11,679	2,647	732.3	27.5688
	2	**187,422,365**	6,715	0	30.9	**187,422,365**	6,715	0	122.5	0.0000
	3	**340,487,089**	8,951	0	3.8	**340,487,089**	8,951	0	237.9	0.0000
	4	**309,586,584**	9,677	0	1.3	**309,586,584**	9,677	0	677.9	0.0000
	5	444,536,794	15,954	5,462	60.3	**182,258,424**	6,541	0	1,443.5	-59.0004
	6	236,240,392	6,246	2,066	0.2	**148,039,222**	3,914	0	124.9	-37.3353
	7	**607,520,858**	13,433	0	15.8	**607,520,858**	13,433	0	916.3	0.0000
	8	241,124,382	12,191	1,407	96.8	**191,042,687**	9,659	2,674	2,260.8	-20.7701
	9	**226,123,995**	4,757	0	0.8	**226,123,995**	4,757	0	221.0	0.0000
	10	**354,815,285**	10,884	3,115	8.9	**354,815,285**	10,884	3,115	470.3	0.0000
Avg.		326,196,579	9,796		27.9	294,800,035	8,621		720.7	-8.9537
8 periods	11	1,550,317,180	16,165	3,310	180.3	**1,482,704,276**	15,460	2,484	4,664.5	-4.3612
	12	**1,625,463,920**	17,980	0	103.7	1,764,776,484	19,521	0	2,880.4	8.5706
	13	**1,102,076,378**	11,453	0	0.7	**1,102,076,378**	11,453	0	627.0	0.0000
	14	1,423,459,632	16,701	0	18.7	**1,360,217,488**	15,959	0	2,970.7	-4.4428
	15	**1,156,701,480**	15,396	0	159.0	1,169,398,450	15,565	0	3,086.2	1.0977
	16	**1,037,649,354**	12,633	0	163.6	1,299,831,032	15,825	2,818	4,514.8	25.2669
	17	**1,236,188,630**	17,285	0	124.9	**1,236,188,630**	17,285	0	3,578.7	0.0000
	18	**1,271,449,952**	15,649	0	61.6	**1,271,449,952**	15,649	0	1,689.7	0.0000
	19	**1,489,848,521**	17,883	2,092	125.9	1,589,990,435	19,085	0	3,001.7	6.7216
	20	**1,464,089,337**	17,855	2,808	63.7	1,555,845,819	18,974	3,207	2,160.4	6.2671
Avg.		1,335,724,438	15,683		104.3	1,364,070,347	16,200		3,001.5	3.6503
12 periods	21	**2,905,035,501**	22,879	2,645	61.3	3,012,458,150	23,725	0	2,638.6	3.6978
	22	**2,526,326,584**	22,230	1,766	181.6	2,592,808,909	22,815	1,766	3,926.3	2.6316
	23	**2,586,793,620**	22,189	0	74.0	2,910,185,329	24,963	1,211	2,340.8	12.5016
	24	**2,745,092,742**	23,139	2,523	73.7	2,753,399,715	23,209	0	2,387.0	0.3026
	25	3,911,466,834	28,039	1,705	135.8	**3,770,293,527**	27,027	0	3,244.8	-3.6092
	26	3,966,384,615	27,042	735	124.7	**3,927,662,020**	26,778	1,130	3,847.7	-0.9763
	27	**3,462,474,633**	26,709	0	240.9	3,711,377,673	28,629	0	2,842.8	7.1886
	28	3,106,309,844	28,536	4,972	159.8	**2,956,637,816**	27,161	0	3,652.1	-4.8183
	29	**2,682,280,094**	19,795	1,791	135.6	2,802,335,761	20,681	1,782	2,796.8	4.4759
	30	3,821,604,621	24,685	3,654	182.3	**3,437,821,791**	22,206	99	1,904.0	-10.0425
Avg.		3,171,376,909	24,524		137.0	3,187,498,069	24,719		2,958.1	1.1352
Avg.		1,611,099,309	16,740		88.4	1,621,848,666	16,606		2,198.7	-1.3022

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 (27), 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,

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	**277,053,250**	8,075	0	994.4	-7.8504
	2	**183,066,191**	6,559	2,058	0.8	187,421,443	6,715	922	441.1	2.3791
	3	340,482,337	8,951	4,752	63.1	**339,152,364**	8,916	3,360	535.5	-0.3906
	4	309,582,278	9,677	4,306	76.5	**277,209,196**	8,665	1,484	781.3	-10.4570
	5	274,293,216	9,844	0	120.1	**182,257,329**	6,541	1,095	2,407.4	-33.5538
	6	181,132,639	4,789	1,708	2.4	**179,167,551**	4,737	0	350.0	-1.0849
	7	**527,061,892**	11,654	1,912	133.9	**527,061,892**	11,654	1,912	1,656.1	0.0000
	8	**166,697,412**	8,428	0	36.9	**166,697,412**	8,428	0	936.7	0.0000
	9	226,123,365	4,757	630	7.3	**226,122,767**	4,757	1,228	429.2	-0.0003
	10	**284,400,266**	8,724	2,134	61.5	**284,400,266**	8,724	2,134	618.2	0.0000
Avg.		279,349,548	8,215		50.3	264,654,347	7,721		915.0	-5.0958
8 periods	11	1,425,351,269	14,862	3,703	246.3	**1,200,933,896**	12,522	1,036	2,632.4	-15.7447
	12	1,492,298,384	16,507	444	301.5	**1,492,297,743**	16,507	1,085	3,898.8	0.0000
	13	1,041,838,902	10,827	0	5.8	**741,805,859**	7,709	375	687.4	-28.7984
	14	**1,151,398,287**	13,509	801	53.7	1,151,398,632	13,509	456	3,121.8	0.0000
	15	1,190,883,867	15,851	1,763	137.6	**1,104,410,088**	14,700	912	3,913.4	-7.2613
	16	**1,037,649,024**	12,633	330	161.1	1,064,753,959	12,963	935	4,125.3	2.6121
	17	1,137,778,191	15,909	1,671	190.0	**1,083,926,464**	15,156	344	6,128.1	-4.7331
	18	1,203,279,118	14,810	3,762	108.9	**1,025,673,954**	12,624	798	3,779.6	-14.7601
	19	**1,111,449,959**	13,341	2,092	193.6	1,257,579,545	15,095	0	3,526.3	13.1477
	20	1,282,624,633	15,642	3,725	126.9	**1,242,694,261**	15,155	584	3,393.1	-3.1132
Avg.		1,207,455,163	14,389		152.5	1,136,547,440	13,594		3,520.6	-5.8651
12 periods	21	2,573,632,748	20,269	3,258	137.4	**2,457,199,628**	19,352	1,220	5,564.3	-4.5241
	22	**2,286,762,128**	20,122	2,562	195.5	2,380,519,253	20,947	2,562	3,351.6	4.1000
	23	2,324,372,040	19,938	0	232.9	**2,259,553,182**	19,382	378	5,180.9	-2.7887
	24	2,704,400,499	22,796	2,961	173.4	**2,517,790,605**	21,223	0	2,540.5	-6.9002
	25	3,310,219,229	23,729	0	188.8	**2,870,233,075**	20,575	0	3,798.6	-13.2918
	26	3,384,818,287	23,077	688	129.9	**3,229,489,415**	22,018	735	4,707.2	-4.5890
	27	**2,952,610,016**	22,776	2,296	308.0	3,214,994,976	24,800	2,624	3,507.9	8.8865
	28	2,991,904,474	27,485	2,686	231.7	**2,717,695,698**	24,966	3,198	5,075.0	-9.1650
	29	2,369,810,419	17,489	1,548	245.5	**2,201,786,731**	16,249	1,516	3,951.7	-7.0902
	30	3,189,962,135	20,605	940	174.1	**2,837,294,505**	18,327	0	3,256.4	-11.0555
Avg.		2,808,849,198	21,829		201.7	2,668,655,707	20,784		4,093.4	-4.6418
Avg.		1,431,884,636	14,811		134.9	1,356,619,165	14,033		2,843.0	-5.2009

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 table, "gap(%)" represents the relative gap between the solutions found by both methods, computed as

$$100\left(\frac{F_{\text{flook}} - F_{\text{cplex}}}{F_{\text{cplex}}}\right)\%, \tag{29}$$

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, 5, 4, and 0 instances

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

Inst.		Myopic approach: Best objective function value	Myopic approach: Objects cost	Myopic approach: Leftovers value	Myopic approach: CPU time	Forward-looking approach: Best objective function value	Forward-looking approach: Objects cost	Forward-looking approach: Leftovers value	Forward-looking approach: CPU time	gap (%)
4 periods	1	**177,005,290**	5,159	0	60.5	277,052,202	8,075	1,048	743.2	56.5220
	2	183,066,047	6,559	2,202	2.6	**165,540,141**	5,931	0	534.1	-9.5735
	3	340,482,702	8,951	4,387	63.4	**205,638,144**	5,406	690	400.9	-39.6039
	4	309,582,332	9,677	4,252	121.9	**187,633,080**	5,865	0	950.3	-39.3915
	5	274,289,096	9,844	4,120	122.4	**182,257,427**	6,541	997	2,654.2	-33.5528
	6	181,132,281	4,789	2,066	2.3	**92,931,111**	2,457	0	215.9	-48.6943
	7	**352,310,540**	7,790	0	135.6	**352,310,540**	7,790	0	1,113.0	0.0000
	8	**166,694,832**	8,428	2,580	96.5	166,694,950	8,428	2,462	1,568.4	0.0001
	9	**226,122,641**	4,757	1,354	8.8	**226,122,641**	4,757	1,354	387.7	0.0000
	10	**178,974,000**	5,490	0	65.3	**178,974,000**	5,490	0	684.5	0.0000
Avg.		238,965,976	7,144		67.9	203,515,424	6,074		925.2	-11.4294
8 periods	11	1,231,334,604	12,839	2,530	150.6	**1,118,166,238**	11,659	1,816	2,543.6	-9.1907
	12	1,661,892,542	18,383	4,190	301.7	**1,459,391,772**	16,143	0	4,490.6	-12.1849
	13	920,593,767	9,567	375	51.8	**776,062,690**	8,065	0	1,226.3	-15.6998
	14	**1,019,203,389**	11,958	867	50.2	1,019,203,408	11,958	848	3,878.2	0.0000
	15	1,190,882,635	15,851	2,995	198.4	**1,048,738,758**	13,959	912	3,914.1	-11.9360
	16	1,210,381,894	14,736	3,674	143.9	**966,517,321**	11,767	525	4,190.4	-20.1477
	17	1,292,683,743	18,075	4,107	242.6	**1,083,926,384**	15,156	424	4,302.7	-16.1491
	18	**911,276,358**	11,216	1,210	173.6	1,025,673,954	12,624	798	4,277.2	12.5536
	19	**1,111,449,683**	13,341	2,368	206.6	1,343,385,248	16,125	4,627	3,499.5	20.8678
	20	1,218,090,995	14,855	4,150	242.6	**1,045,977,464**	12,756	1,780	3,820.2	-14.1298
Avg.		1,176,778,961	14,082		176.2	1,088,704,324	13,021		3,614.3	-6.6017
12 periods	21	2,263,564,302	17,827	1,196	174.9	**2,177,222,273**	17,147	905	4,195.6	-3.8144
	22	2,254,372,691	19,837	3,174	225.2	**2,151,866,309**	18,935	1,766	7,007.4	-4.5470
	23	**2,093,542,769**	17,958	871	182.2	2,198,114,815	18,855	1,085	4,105.2	4.9950
	24	2,704,399,467	22,796	3,993	192.5	**2,198,543,471**	18,532	349	2,637.3	-18.7049
	25	3,374,945,006	24,193	2,687	209.1	**2,750,262,215**	19,715	0	4,346.5	-18.5094
	26	2,790,050,551	19,022	1,299	218.3	**2,658,923,500**	18,128	900	3,584.4	-4.6998
	27	**2,719,263,555**	20,976	2,157	312.6	2,804,696,495	21,635	0	4,507.4	3.1418
	28	2,947,923,389	27,081	5,947	329.4	**2,331,585,459**	21,419	1,205	4,223.7	-20.9075
	29	2,280,785,228	16,832	1,268	247.8	**2,163,304,424**	15,965	971	6,646.0	-5.1509
	30	2,677,059,585	17,292	1,395	244.3	**2,546,550,563**	16,449	1,372	3,181.9	-4.8751
Avg.		2,610,590,654	20,381		233.6	2,398,106,952	18,678		4,443.5	-7.3072
Avg.		1,342,111,864	13,869		159.3	1,230,108,900	12,591		2,994.3	-8.4461

(out of 10) respectively; while the average gap (29) between CPLEX and the forward-looking approach was 5.8%, 13.4%, -1.1%, and -4.7%. 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 2 solutions (one of them known to be optimal) and improved other 2 solutions. For the instances with $\xi = 3$, the forward-looking approach matched 3 solutions (known to be optimal) and improved other 3. For the instances with $\xi = 4$, the forward-looking approach improved 5 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

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	68.2	277,048,397	8,075	4,853	1,045.8	56.5216
	2	183,066,038	6,559	2,211	2.6	**165,538,679**	5,931	1,462	508.7	-9.5743
	3	340,482,702	8,951	4,387	63.4	**205,637,388**	5,406	1,446	366.0	-39.6042
	4	309,582,269	9,677	4,315	122.0	**309,582,225**	9,677	4,359	872.1	0.0000
	5	274,288,961	9,844	4,255	123.0	**182,257,457**	6,541	967	1,686.5	-33.5528
	6	181,131,635	4,789	2,712	2.5	**92,930,797**	2,457	314	353.7	-48.6943
	7	**352,308,306**	7,790	2,234	193.9	352,308,700	7,790	1,840	1,553.6	0.0001
	8	**166,694,901**	8,428	2,511	96.9	166,694,948	8,428	2,464	1,641.1	0.0000
	9	**226,122,426**	4,757	1,569	8.9	**226,122,426**	4,757	1,569	470.5	0.0000
	10	178,973,172	5,490	828	65.3	**178,972,975**	5,490	1,025	669.4	-0.0001
Avg.		238,965,369	7,144		74.7	215,709,399	6,455		916.7	-7.4904
8 periods	11	**997,133,908**	10,397	774	80.5	1,007,107,737	10,501	1,169	3,315.6	1.0002
	12	1,555,759,725	17,209	2,711	307.0	**1,283,103,972**	14,193	0	6,345.4	-17.5256
	13	1,006,137,497	10,456	1,559	60.9	**741,805,436**	7,709	798	1,203.2	-26.2720
	14	**1,019,202,506**	11,958	1,750	210.5	1,019,203,304	11,958	952	4,411.3	0.0001
	15	1,190,882,363	15,851	3,267	185.6	**1,010,121,118**	13,445	1,732	5,956.5	-15.1788
	16	1,210,381,894	14,736	3,674	201.3	**1,037,648,275**	12,633	1,079	2,826.9	-14.2710
	17	1,137,777,475	15,909	2,387	288.6	**1,031,360,898**	14,421	180	5,989.4	-9.3530
	18	1,203,278,753	14,810	4,127	188.2	**1,025,673,270**	12,624	1,482	6,018.4	-14.7601
	19	1,111,449,235	13,341	2,816	208.1	**1,026,389,387**	12,320	2,133	5,096.1	-7.6531
	20	1,282,623,697	15,642	4,661	308.5	**1,049,996,019**	12,805	1,176	4,192.9	-18.1369
Avg.		1,171,462,705	14,031		203.9	1,023,240,942	12,261		4,535.6	-12.2150
12 periods	21	**2,197,791,998**	17,309	968	226.6	2,243,630,156	17,670	424	3,523.3	2.0856
	22	2,254,372,691	19,837	3,174	199.3	**1,940,033,795**	17,071	0	4,644.8	-13.9435
	23	**2,061,483,504**	17,683	636	223.2	2,073,956,989	17,790	1,211	5,502.0	0.6051
	24	2,301,874,270	19,403	635	189.4	**2,173,748,840**	18,323	265	2,756.2	-5.5661
	25	2,981,413,301	21,372	2,071	141.9	**2,779,137,217**	19,922	1,705	4,666.2	-6.7846
	26	2,929,977,991	19,976	1,809	253.6	**2,658,922,840**	18,128	1,560	4,117.1	-9.2511
	27	**2,727,819,075**	21,042	2,679	364.1	2,727,819,151	21,042	2,603	7,119.4	0.0000
	28	2,792,803,602	25,656	5,934	337.3	**2,421,391,653**	22,244	1,211	4,623.5	-13.2989
	29	2,491,626,343	18,388	2,821	178.7	**2,147,856,651**	15,851	1,402	5,261.6	-13.7970
	30	2,677,058,982	17,292	1,998	245.5	**2,262,001,370**	14,611	595	4,625.8	-15.5042
Avg.		2,541,622,176	19,796		236.0	2,342,849,866	18,265		4,684.0	-7.5455
Avg.		1,317,350,083	13,657		171.5	1,193,933,402	12,327		3,378.8	-9.0836

this set, the cases in which CPLEX wins are concentrated in the instances with $\xi = 1, 2$, 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 16 out of the 20 instances with $\xi = 4$, CPLEX was able to find a feasible solution;

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

Inst.		Myopic approach: Best objective function value	Objects cost	Leftovers value	CPU time	Forward-looking approach: Best objective function value	Objects cost	Leftovers value	CPU time	gap (%)
8 periods	11	1,215,891,809	12,678	4,459	189.5	**909,955,304**	9,488	824	4,170.5	-25.1615
	12	1,555,758,322	17,209	4,114	306.5	**1,254,444,657**	13,876	1,247	5,958.0	-19.3676
	13	773,366,591	8,037	1,771	68.5	**594,579,167**	6,179	1,287	1,887.5	-23.1181
	14	1,019,201,343	11,958	2,913	206.2	**900,474,723**	10,565	1,357	4,569.7	-11.6490
	15	1,190,882,133	15,851	3,497	156.2	**1,003,810,128**	13,361	1,802	5,501.9	-15.7087
	16	1,210,381,894	14,736	3,674	201.4	**980,726,922**	11,940	798	4,128.7	-18.9738
	17	1,137,777,262	15,909	2,600	288.4	**1,025,352,729**	14,337	837	6,173.7	-9.8811
	18	1,203,277,781	14,810	5,099	188.3	**925,900,439**	11,396	1,769	3,356.6	-23.0518
	19	1,111,448,881	13,341	3,170	268.7	**883,095,903**	10,600	697	8,360.2	-20.5455
	20	1,190,621,519	14,520	3,961	305.4	**873,944,862**	10,658	480	6,159.4	-26.5976
Avg.		1,160,860,754	13,905		217.9	935,228,483	11,240		5,026.6	-19.4055
12 periods	21	1,983,206,578	15,619	328	173.5	**1,873,119,997**	14,752	451	6,038.4	-5.5509
	22	1,813,886,558	15,961	1,287	262.2	**1,727,516,865**	15,201	780	9,290.1	-4.7616
	23	1,741,938,045	14,942	315	250.5	**1,691,575,639**	14,510	161	8,961.1	-2.8912
	24	2,301,871,943	19,403	2,962	187.4	**1,969,220,958**	16,599	1,407	3,904.1	-14.4513
	25	2,883,203,059	20,668	3,609	202.4	**2,434,989,295**	17,455	660	6,673.6	-15.5457
	26	2,790,048,502	19,022	3,348	193.4	**2,290,036,133**	15,613	642	8,996.0	-17.9213
	27	2,727,820,154	21,042	1,600	247.7	**2,391,282,662**	18,446	1,440	5,000.3	-12.3372
	28	2,303,933,956	21,165	3,284	309.0	**2,039,308,091**	18,734	213	11,452.9	-11.4858
	29	1,989,452,967	14,682	2,079	160.0	**1,970,076,007**	14,539	2,110	5,093.2	-0.9740
	30	2,677,058,826	17,292	2,154	244.8	**2,153,321,736**	13,909	99	5,106.9	-19.5639
Avg.		2,321,242,059	17,980		223.1	2,054,044,738	15,976		7,051.6	-10.5483
Avg.		1,741,051,406	15,942		220.5	1,494,636,611	13,608		6,039.1	-14.9769

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	6/3/1	-5.01	5/3/2	-11.43	6/1/3	-7.49	–	–
8	2/3/5	3.65	7/0/3	-5.87	7/0/3	-6.60	8/0/2	-12.22	10/0/0	-19.41
12	4/0/6	1.14	8/0/2	-4.64	8/0/2	-7.31	7/0/3	-7.55	10/0/0	-10.55
Avg.	9/9/12	-1.30	21/3/6	-5.20	20/3/7	-8.45	21/1/8	-9.08	20/0/0	-14.98

while it failed to find a feasible solution in the other 4 instances. Of those 16 instances, the forward-looking approach found better solutions in 15 instances, with an average gap of -33.62%. Of the total 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 -74.81%.

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 $\xi \in \{1, 2, 3, 4\}$.

		CPLEX						Forward-looking approach				
ξ	Inst.	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	314,108,050	**314,108,050**	9,155	0	0.0000	0.2	400,703,843	11,679	2,647	732.3	27.5688
	2	183,065,474	**187,422,365**	6,715	0	2.3246	7,200.0	**187,422,365**	6,715	0	122.5	0.0000
	3	339,152,904	**339,152,904**	8,916	2,820	0.0000	0.3	340,487,089	8,951	0	237.9	0.3934
	4	309,586,584	**309,586,584**	9,677	0	0.0000	0.2	**309,586,584**	9,677	0	677.9	0.0000
	5	182,258,424	**182,258,424**	6,541	0	0.0000	50.6	**182,258,424**	6,541	0	1,443.5	0.0000
1	6	148,039,222	**148,039,222**	3,914	0	0.0000	0.1	**148,039,222**	3,914	0	124.9	0.0000
	7	580,789,740	**580,790,380**	12,842	1,912	0.0001	7,200.0	607,520,858	13,433	0	916.3	4.6024
	8	80,065,392	**186,634,644**	9,436	0	57.1005	7,200.0	191,042,687	9,659	2,674	2,260.8	2.3619
	9	226,123,995	**226,123,995**	4,757	0	0.0000	0.2	**226,123,995**	4,757	0	221.0	0.0000
	10	288,510,000	**288,510,000**	8,850	0	0.0000	268.4	354,815,285	10,884	3,115	470.3	22.9820
	Avg.		276,262,657	8,080		5.9425	2,192.0	294,800,035	8,621		720.7	5.7909
	1	277,053,250	**277,053,250**	8,075	0	0.0000	0.3	**277,053,250**	8,075	0	994.4	0.0000
	2	125,208,746	**125,208,746**	4,486	0	0.0000	1,942.0	187,421,443	6,715	922	441.1	49.6872
	3	205,638,834	**205,638,834**	5,406	0	0.0000	0.3	339,152,364	8,916	3,360	535.5	64.9262
	4	216,808,300	**277,209,196**	8,665	1,484	21.7889	7,200.0	**277,209,196**	8,665	1,484	781.3	0.0000
	5	162,301,312	235,866,007	8,465	2,753	31.1892	7,200.0	**182,257,329**	6,541	1,095	2,407.4	-22.7284
2	6	136,049,331	**136,049,331**	3,597	0	0.0000	1.8	179,167,551	4,737	0	350.0	31.6931
	7	406,039,028	**491,516,168**	10,868	0	17.3905	7,200.0	527,061,892	11,654	1,912	1,656.1	7.2319
	8	80,062,469	186,631,619	9,436	3,025	57.1013	7,200.0	**166,697,412**	8,428	0	936.7	-10.6810
	9	226,117,517	**226,122,466**	4,757	1,529	0.0022	7,200.0	226,122,767	4,757	1,228	429.2	0.0001
	10	249,388,985	**249,388,985**	7,650	1,015	0.0000	551.8	284,400,266	8,724	2,134	618.2	14.0388
	Avg.		241,068,460	7,141		12.7472	3,849.6	264,654,347	7,721		915.0	13.4168
	1	177,005,290	**177,005,290**	5,159	0	0.0000	4.5	277,052,202	8,075	1,048	743.2	56.5226
	2	111,055,089	**165,538,722**	5,931	1,419	32.9129	7,200.0	165,540,141	5,931	0	534.1	0.0000
	3	115,486,404	**205,637,382**	5,406	1,452	43.8398	7,200.0	205,638,144	5,406	690	400.9	0.0000
	4	127,232,184	309,582,248	9,677	4,336	58.9020	7,200.0	**187,633,080**	5,865	0	950.3	-39.3924
	5	73,560,960	203,212,152	7,293	0	63.8009	7,200.0	**182,257,427**	6,541	997	2,654.2	-10.3113
3	6	92,931,111	**92,931,111**	2,457	0	0.0000	44.0	**92,931,111**	2,457	0	215.9	0.0000
	7	352,310,540	**352,310,540**	7,790	0	0.0000	6.3	**352,310,540**	7,790	0	1,113.0	0.0000
	8	36,551,592	203,244,701	10,276	4,303	82.0160	7,200.0	**166,694,950**	8,428	2,462	1,568.4	-17.9837
	9	226,118,625	**226,122,492**	4,757	1,503	0.0017	7,200.0	226,122,641	4,757	1,354	387.7	0.0000
	10	178,974,000	**178,974,000**	5,490	0	0.0000	9.9	**178,974,000**	5,490	0	684.5	0.0000
	Avg.		211,455,864	6,424		28.1473	4,326.5	203,515,424	6,074		925.2	-1.1165
	1	176,987,996	**177,003,339**	5,159	1,951	0.0087	7,200.0	277,048,397	8,075	4,853	1,045.8	56.5216
	2	111,048,262	169,836,152	6,085	2,283	34.6145	7,200.0	**165,538,679**	5,931	1,462	508.7	-2.5304
	3	115,477,259	**205,637,085**	5,406	1,749	43.8441	7,200.0	205,637,388	5,406	1,446	366.0	0.0001
	4	127,219,757	314,860,300	9,842	4,964	59.5949	7,200.0	**309,582,225**	9,677	4,359	872.1	-1.6763
	5	53,604,707	276,768,471	9,933	4,641	80.6319	7,200.0	**182,257,457**	6,541	967	1,686.5	-34.1480
4	6	92,925,615	**92,930,733**	2,457	378	0.0055	7,200.0	92,930,797	2,457	314	353.7	0.0001
	7	352,266,598	406,035,779	8,978	3,249	13.2425	7,200.0	**352,308,700**	7,790	1,840	1,553.6	-13.2321
	8	36,542,003	347,683,703	17,579	11,338	89.4899	7,200.0	**166,694,948**	8,428	2,464	1,641.1	-52.0556
	9	226,115,028	**226,122,389**	4,757	1,606	0.0033	7,200.0	226,122,426	4,757	1,569	470.5	0.0000
	10	178,945,728	**178,972,785**	5,490	1,215	0.0151	7,200.0	178,972,975	5,490	1,025	669.4	0.0001
	Avg.		239,585,074	7,569		32.1450	7,200.0	215,709,399	6,455		916.7	-4.7121

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

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 $\xi \in \{4, P\}$.

		CPLEX						Forward-looking approach				
ξ	Inst.	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 (%)
4	11	473,194,584	1,693,795,866	17,661	0	72.0631	7,200.0	**1,007,107,737**	10,501	1,169	3,315.6	-40.5414
	12	436,284,255	2,225,925,343	24,622	1,945	80.3999	7,200.0	**1,283,103,972**	14,193	0	6,345.4	-42.3564
	13	372,869,732	**612,863,014**	6,369	380	39.1594	7,200.0	741,805,436	7,709	798	1,203.2	21.0394
	14	222,188,546	1,658,613,465	19,460	1,255	86.6040	7,200.0	**1,019,203,304**	11,958	952	4,411.3	-38.5509
	15	262,867,798	2,360,960,250	31,425	0	88.8661	7,200.0	**1,010,121,118**	13,445	1,732	5,956.5	-57.2157
	16	383,736,775	2,283,682,476	27,803	338	83.1966	7,200.0	**1,037,648,275**	12,633	1,079	2,826.9	-54.5625
	17	Solution not found					7,200.0	**1,031,360,898**	14,421	180	5,989.4	–
	18	441,237,340	1,205,070,336	14,832	0	63.3849	7,200.0	**1,025,673,270**	12,624	1,482	6,018.4	-14.8869
	19	618,229,016	1,814,594,666	21,781	2,225	65.9302	7,200.0	**1,026,389,387**	12,320	2,133	5,096.1	-43.4370
	20	538,241,436	1,151,593,956	14,044	0	53.2612	7,200.0	**1,049,996,019**	12,805	1,176	4,192.9	-8.8224
	Avg.		1,667,455,486	19,777		70.3184	7,200.0	1,023,240,942	12,261		4,535.6	-31.0371
	21	997,752,674	2,977,411,727	23,449	1,599	66.4893	7,200.0	**2,243,630,156**	17,670	424	3,523.3	-24.6449
	22	848,244,208	3,940,979,158	34,678	2,152	78.4763	7,200.0	**1,940,033,795**	17,071	0	4,644.8	-50.7728
	23	1,113,793,829	2,572,336,398	22,065	1,302	56.7011	7,200.0	**2,073,956,989**	17,790	1,211	5,502.0	-19.3746
	24	992,737,680	3,628,332,840	30,584	0	72.6393	7,200.0	**2,173,748,840**	18,323	265	2,756.2	-40.0896
	25	664,024,760	5,652,018,841	40,516	3,675	88.2515	7,200.0	**2,779,137,217**	19,922	1,705	4,666.2	-50.8293
	26	930,198,690	4,379,714,810	29,860	690	78.7612	7,200.0	**2,658,922,840**	18,128	1,560	4,117.1	-39.2900
	27	Solution not found					7,200.0	**2,727,819,151**	21,042	2,603	7,119.4	–
	28	Solution not found					7,200.0	**2,421,391,653**	22,244	1,211	4,623.5	–
	29	Solution not found					7,200.0	**2,147,856,651**	15,851	1,402	5,261.6	–
	30	Solution not found					7,200.0	**2,262,001,370**	14,611	595	4,625.8	–
	Avg.		3,858,465,629	30,192		73.5531	7,200.0	2,342,849,866	18,265		4,684.0	-37.5002
P	11	Solution not found					7,200.0	**909,955,304**	9,488	824	4,170.5	–
	12	Solution not found					7,200.0	**1,254,444,657**	13,876	1,247	5,958.0	–
	13	191,248,610	2,094,633,046	21,768	14,522	90.8700	7,200.0	**594,579,167**	6,179	1,287	1,887.5	-71.6100
	14	Solution not found					7,200.0	**900,474,723**	10,565	1,357	4,569.7	–
	15	Solution not found					7,200.0	**1,003,810,128**	13,361	1,802	5,501.9	–
	16	Solution not found					7,200.0	**980,726,922**	11,940	798	4,128.7	–
	17	Solution not found					7,200.0	**1,025,352,729**	14,337	837	6,173.7	–
	18	303,165,605	4,212,836,468	51,852	34,828	92.8000	7,200.0	**925,900,439**	11,396	1,769	3,356.6	-78.0200
	19	Solution not found					7,200.0	**883,095,903**	10,600	697	8,360.2	–
	20	Solution not found					7,200.0	**873,944,862**	10,658	480	6,159.4	–
	Avg.		3,153,734,757	36,810		91.8350	7,200.0	935,228,483	11,240		5,026.6	-74.8150
	21	Solution not found					7,200.0	**1,873,119,997**	14,752	451	6,038.4	–
	22	Solution not found					7,200.0	**1,727,516,865**	15,201	780	9,290.1	–
	23	Solution not found					7,200.0	**1,691,575,639**	14,510	161	8,961.1	–
	24	Solution not found					7,200.0	**1,969,220,958**	16,599	1,407	3,904.1	–
	25	Solution not found					7,200.0	**2,434,989,295**	17,455	660	6,673.6	–
	26	Solution not found					7,200.0	**2,290,036,133**	15,613	642	8,996.0	–
	27	Solution not found					7,200.0	**2,391,282,662**	18,446	1,440	5,000.3	–
	28	Solution not found					7,200.0	**2,039,308,091**	18,734	213	11,452.9	–
	29	Solution not found					7,200.0	**1,970,076,007**	14,539	2,110	5,093.2	–
	30	Solution not found					7,200.0	**2,153,321,736**	13,909	99	5,106.9	–
	Avg.		–	–		–	7,200.0	2,054,044,738	15,976		7,051.7	–

the period. On the other and, 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.

In the present work, as well as in Birgin et al. (2020), single-period subproblems are solved with an exact commercial solver. On the one hand, the proposed method can be applied to instances with a large number of periods. On the other hand, solving the single-period subproblems exactly limits the applicability to instances with larger 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.

References

Ali, R., Muhammad, S., and Takahashi, R. H. C. (2021). Decision making viva genetic algorithm for the utilization of leftovers. *International Journal of Intelligent Systems*, 36:1746–1769.

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

Andrade, R., Birgin, E. G., Morabito, R., and 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.

Baykasoglu, A. and 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*, 128:105173.

Birgin, E. G., Romão, O. C., and Ronconi, D. P. (2020). The multiperiod two-dimensional non-guillotine cutting stock problem with usable leftovers. *International Transactions in Operational Research*, 27:1392–1418.

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

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

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

do Nascimento, D. N., de Araujo, S. A., and Cherri, A. C. (2021). Integrated lot-sizing and one-dimensional cutting stock problem with usable leftovers. *Annals of Operations Research*, to appear.

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

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

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

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

Silva, E., Avelos, F., and Valério de Carvalho, J. M. (2010). An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, 205:699–708.

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

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

Viegas, J. L., Vieira, S. M., Henriques, E. M. P., and 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.

Appendix

Table 10 describes in detail the thirty instances with four, eight, and twelve periods considered in the present work. Instances were generated with the random instances generator introduced in Birgin et al. (2020), where additional twenty five instances with four periods are also described. The number of binary variables, continuous variables, and constraints of each instance, for $\xi \in \{0, 1, 2, 3, 4\}$ is given in Table 1. The random instances generator is available at `https://github.com/oberlan/bromro2`.

Table 10: Description of the considered thirty instances with four, eight, and twelve periods.

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

Continued on next page

Table 10: – continued from previous page

Inst.	P	Objects		Items			
		m^s	$W_j^s \times H_j^s$	n^s	$\tilde{n}^s$	d	$w_i^s \times h_i^s$
12	8	3	68 × 37, 70 × 43, 97 × 52	7	5	3	20 × 14, 14 × 10, 20 × 15, 3(17 × 19), 7 × 13
		3	88 × 39, 89 × 35, 55 × 79	8	4		3(7 × 17), 3(15 × 11), 10 × 12, 20 × 10
		2	66 × 77, 58 × 88	11	5		18 × 9, 3(10 × 20), 2(18 × 5), 2(7 × 12), 3(14 × 15)
		2	95 × 69, 85 × 97	8	4		2(20 × 14), 14 × 18, 3(8 × 17), 2(14 × 15)
		2	30 × 84, 65 × 56	6	3		3(5 × 20), 2(12 × 13), 14 × 9
		3	75 × 63, 42 × 55, 73 × 89	5	3		$\underline{5 \times 9}$, 2(17 × 15), 2(11 × 9)
		3	90 × 57, 67 × 52, 76 × 86	10	4		3(20 × 15), 13 × 19, 3($\underline{16 \times 5}$), 3(19 × 5)
		2	46 × 91, 88 × 56	10	5		2(10 × 18), 14 × 9, 3(11 × 17), 3(17 × 9), $\underline{9 \times 8}$
13	8	2	58 × 43, 39 × 51	5	3	4	10 × 18, 3(9 × 9), 12 × 8
		3	94 × 47, 97 × 39, 85 × 70	6	3		8 × 8, 3(17 × 6), 2(15 × 6)
		2	84 × 72, 85 × 77	6	3		13 × 18, 3(17 × 6), 2(5 × 13)
		3	83 × 81, 55 × 67, 81 × 86	7	4		12 × 12, 3($\underline{13 \times 5}$), 15 × 11, 2($\underline{5 \times 9}$)
		3	51 × 61, 97 × 53, 41 × 46	2	2		18 × 14, $\underline{6 \times 8}$
		2	62 × 45, 60 × 75	3	2		2(6 × 19), 6 × 16
		3	44 × 91, 70 × 99, 30 × 51	3	2		10 × 9, 2($\underline{11 \times 7}$)
		3	96 × 85, 41 × 59, 98 × 73	5	2		3(18 × 9), 2(20 × 8)
14	8	3	33 × 32, 57 × 91, 62 × 84	4	2	1	3(12 × 13), 10 × 10
		2	91 × 83, 81 × 68	4	3		2(16 × 18), 16 × 7, 15 × 8
		2	70 × 35, 39 × 72	7	4		8 × 19, 2(10 × 10), 3(6 × 16), 10 × 6
		2	78 × 92, 51 × 93	5	3		20 × 14, 3(15 × 8), 16 × 17
		3	50 × 70, 71 × 81, 33 × 47	10	5		2(18 × 5), 13 × 15, 2(15 × 5), 3(17 × 5), 2(15 × 17)
		3	57 × 50, 34 × 86, 94 × 45	8	4		3(19 × 16), 3(18 × 12), 14 × 14, 14 × 17
		3	68 × 94, 50 × 68, 48 × 53	11	5		3($\underline{5 \times 5}$), 3(8 × 16), 3(14 × 12), 16 × 20, 11 × 6
		2	61 × 64, 73 × 89	6	3		3(7 × 6), 2(12 × 15), 16 × 5
15	8	2	85 × 40, 55 × 36	5	2	4	3(17 × 13), 2($\underline{8 \times 11}$)
		3	59 × 53, 92 × 88, 51 × 58	10	5		2(18 × 12), 3($\underline{7 \times 18}$), 3(11 × 17), 13 × 10, $\underline{8 \times 11}$
		3	98 × 82, 2(44 × 49)	6	4		16 × 6, 3(18 × 17), 19 × 19, 19 × 16
		2	51 × 89, 32 × 70	4	2		3(10 × 17), 13 × 20
		4	35 × 51, 38 × 80, 2(31 × 49)	7	3		3(18 × 14), 2(15 × 8), 2($\underline{13 \times 5}$)
		3	67 × 77, 37 × 55, 39 × 78	8	3		2(17 × 6), 3($\underline{10 \times 6}$), 3(16 × 17)
		2	88 × 70, 54 × 83	11	5		8 × 20, 2(11 × 11), 3(11 × 16), 2(15 × 10), 3(20 × 17)
		2	57 × 83, 45 × 66	6	4		14 × 14, 3(16 × 10), 14 × 20, 10 × 7
16	8	5	31 × 98, 2(51 × 39), 2(30 × 64)	5	3	2	2(20 × 20), 2(20 × 17), 18 × 14
		2	86 × 87, 82 × 98	4	2		3(7 × 9), 13 × 5
		3	68 × 97, 65 × 65, 78 × 34	10	5		2(17 × 13), 3(16 × 12), 12 × 11, 6 × 17, 3($\underline{7 \times 5}$)
		2	54 × 85, 53 × 59	4	2		12 × 6, 3(7 × 11)
		2	43 × 64, 35 × 85	9	3		3(14 × 9), 3(16 × 17), 3(15 × 18)
		3	82 × 99, 38 × 98, 52 × 53	13	5		3($\underline{7 \times 5}$), 3(9 × 10), 3(15 × 7), 13 × 10, 3($\underline{6 \times 6}$)
		4	66 × 47, 3(35 × 41)	6	3		20 × 7, 2(19 × 12), 3(20 × 18)
		2	73 × 50, 38 × 84	3	2		14 × 19, 2(17 × 11)
17	8	2	81 × 37, 33 × 64	6	3	2	2(9 × 15), 19 × 18, 3(11 × 14)
		3	34 × 83, 59 × 86, 72 × 44	5	3		20 × 15, 14 × 10, 3(18 × 14)
		2	55 × 91, 32 × 43	8	4		17 × 7, 2(14 × 20), 2(8 × 7), 3(8 × 18)
		2	41 × 96, 41 × 86	7	5		2(9 × 9), 18 × 7, 15 × 16, 17 × 18, 2(8 × 15)
		2	80 × 86, 74 × 59	11	4		3(14 × 14), 3(6 × 20), 3(19 × 8), 2(11 × 12)
		4	85 × 39, 85 × 63, 2(51 × 35)	10	4		2(20 × 16), 3(14 × 10), 2(18 × 20), 3(8 × 17)
		2	78 × 53, 62 × 93	9	5		3(20 × 16), 2(11 × 5), 2(15 × 12), 14 × 14, 9 × 14
		2	56 × 66, 52 × 85	15	5		3($\underline{6 \times 8}$), 3($\underline{8 \times 5}$), 3(11 × 17), 3(12 × 16), 3(20 × 6)

Continued on next page

Table 10: – continued from previous page

Inst.	P	Objects		Items			
		m^s	$W_j^s \times H_j^s$	n^s	$\tilde{n}^s$	d	$w_i^s \times h_i^s$
18	8	2	45 × 83, 97 × 52	7	3	2	15 × 15, 3(11 × 13), 3(18 × 13)
		2	89 × 87, 88 × 45	8	5		2(18 × 9), 6 × 7, 2(12 × 8), 8 × 19, 2($\underline{18 \times 6}$)
		3	2(65 × 33), 92 × 72	8	3		3(19 × 20), 2(15 × 14), 3(9 × 14)
		3	76 × 40, 54 × 71, 43 × 78	9	4		3(7 × 8), 5 × 17, 3(6 × 11), 2(17 × 15)
		2	72 × 74, 89 × 73	5	2		3(11 × 7), 2(20 × 16)
		4	59 × 38, 2(44 × 32), 46 × 47	7	4		6 × 17, 2(18 × 16), 2(8 × 15), 2(18 × 11)
		3	56 × 41, 100 × 45, 40 × 92	2	2		13 × 20, 18 × 13
		2	73 × 77, 83 × 54	6	3		2($\underline{5 \times 7}$), 2(16 × 18), 2(10 × 9)
19	8	2	78 × 86, 72 × 67	10	5	2	3($\underline{15 \times 5}$), 3($\underline{6 \times 6}$), 18 × 10, 2(8 × 10), 14 × 19
		3	53 × 67, 37 × 80, 67 × 56	8	4		2(17 × 5), 2(20 × 15), 2(15 × 13), 2(15 × 9)
		3	57 × 85, 52 × 50, 75 × 37	6	3		2(17 × 9), 2(9 × 9), 2(12 × 14)
		3	64 × 44, 45 × 96, 75 × 52	10	5		3(18 × 20), 2(13 × 9), 8 × 9, 9 × 7, 3(14 × 14)
		2	56 × 93, 53 × 49	9	4		3(16 × 10), 3(10 × 14), 12 × 17, 2(6 × 15)
		2	51 × 89, 65 × 72	5	3		16 × 14, 18 × 8, 3(16 × 5)
		2	92 × 64, 81 × 95	6	3		3(19 × 7), 2(6 × 14), 17 × 16
		3	62 × 52, 32 × 97, 95 × 35	8	4		3(7 × 16), 2(10 × 14), 11 × 12, 2(13 × 8)
20	8	3	75 × 82, 69 × 79, 76 × 64	5	3	3	2(14 × 10), 2(15 × 13), 14 × 12
		2	49 × 68, 61 × 79	12	5		3(11 × 18), 2(6 × 12), 2($\underline{7 \times 7}$), 3($\underline{5 \times 12}$), 2(13 × 18)
		3	92 × 41, 74 × 51, 78 × 93	7	4		$\underline{10 \times 5}$, 2(13 × 6), 2(8 × 10), 2(5 × 13)
		3	61 × 85, 45 × 51, 34 × 50	7	3		3(8 × 19), 14 × 10, 3(9 × 11)
		2	41 × 50, 63 × 84	6	3		2(13 × 20), 2(18 × 12), 2($\underline{10 \times 5}$)
		2	81 × 43, 53 × 45	6	4		2(7 × 14), 13 × 7, 2(9 × 11), 19 × 17
		3	35 × 82, 2(33 × 34)	7	5		6 × 14, 2(17 × 19), 19 × 10, 2(15 × 9), 11 × 11
		3	92 × 52, 83 × 65, 70 × 70	13	5		3(13 × 18), 2(16 × 6), 3(12 × 8), 3(5 × 18), 2(19 × 11)
21	12	2	65 × 50, 93 × 92	5	2	2	2(7 × 8), 3(12 × 10)
		2	90 × 68, 57 × 69	7	4		2(13 × 6), 3(19 × 14), 6 × 11, $\underline{6 \times 5}$
		3	78 × 71, 56 × 70, 62 × 100	6	3		19 × 15, 2(8 × 17), 3(15 × 19)
		2	50 × 84, 30 × 49	7	4		2(7 × 7), 14 × 17, 3(14 × 13), 8 × 16
		2	73 × 99, 44 × 72	4	2		7 × 13, 3(8 × 7)
		3	48 × 50, 70 × 79, 100 × 52	10	5		17 × 16, 2(13 × 17), 2($\underline{5 \times 10}$), 2(16 × 12), 3(6 × 15)
		3	36 × 93, 36 × 77, 92 × 90	4	2		2(13 × 15), 2(9 × 18)
		3	74 × 65, 47 × 70, 100 × 34	4	2		3(15 × 18), 11 × 9
		2	50 × 81, 70 × 87	5	3		16 × 10, 2(16 × 17), 2(10 × 13)
		2	52 × 86, 46 × 48	9	5		11 × 14, 2(19 × 8), 7 × 14, 2(15 × 6), 3(15 × 19)
		2	93 × 47, 31 × 89	5	3		13 × 16, 15 × 18, 3(18 × 7)
		2	81 × 92, 37 × 80	11	5		3(9 × 14), 2(16 × 8), 2(5 × 19), 15 × 7, 3(14 × 17)
22	12	2	73 × 35, 72 × 91	5	4	2	$\underline{6 \times 5}$, 7 × 8, 16 × 12, 2(11 × 8)
		2	39 × 63, 54 × 63	8	3		2(13 × 13), 3(7 × 19), 3(11 × 7)
		3	96 × 44, 63 × 56, 54 × 53	5	4		8 × 20, 15 × 11, 18 × 8, 2(14 × 9)
		2	45 × 82, 69 × 37	12	5		3(17 × 17), 3(19 × 11), 13 × 11, 3(9 × 11), 2(7 × 14)
		3	72 × 62, 63 × 36, 37 × 97	5	4		18 × 13, 19 × 15, 2(18 × 19), 15 × 14
		2	39 × 37, 84 × 42	6	2		3(17 × 6), 3(10 × 5)
		3	2(31 × 38), 98 × 38	13	5		2(8 × 18), 3(8 × 16), 3(6 × 13), 2(16 × 7), 3(8 × 7)
		3	99 × 67, 94 × 93, 65 × 87	12	5		3(14 × 6), 20 × 19, 2(20 × 14), 3(17 × 17), 3(12 × 14)
		2	78 × 66, 42 × 95	6	3		3(9 × 5), 18 × 13, 2($\underline{6 \times 5}$)
		2	78 × 50, 84 × 44	6	3		3(13 × 13), 12 × 9, 2(15 × 16)
		3	76 × 51, 70 × 88, 76 × 57	6	2		3(15 × 12), 3(7 × 12)
		3	71 × 40, 44 × 52, 55 × 58	6	3		$\underline{5 \times 18}$, 3(12 × 6), 2(6 × 17)

Continued on next page

Table 10: – continued from previous page

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

Continued on next page

Table 10: – continued from previous page

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