Noname manuscript No.
(will be inserted by the editor)

Integrated lot-sizing and one-dimensional cutting stock problem with usable leftovers

D. N. do Nascimento · S. A. de Araujo · A. C. Cherri

Received: date / Accepted: date

Abstract This paper addresses the integration of the lot-sizing problem and the one-dimensional cutting stock problem with usable leftovers (LSP-CSPUL). This integration aims to minimize the cost of cutting items from objects available in stock, allowing the bringing forward production of items that have known demands in a future planning horizon. The generation of leftovers, that will be used to cut future items, is also allowed and these leftovers are not considered waste in the current period. Inventory costs for items and leftovers are also considered. A mathematical model for the LSP-CSPUL is proposed to represent this problem and an approach, using the simplex method with column generation, is proposed to solve the linear relaxation of this model. A heuristic procedure, based on a *relax-and-fix* strategy, was also proposed to find integer solutions. Computational tests were performed and the results show the contributions of the proposed mathematical model, as well as, the quality of the solutions obtained using the proposed method.

Keywords Lot-sizing problem · Cutting stock problem · Usable leftovers · Mathematical modeling · Heuristic procedure

1 Introduction

The Cutting Stock Problem (CSP) consists of cutting large objects available in stock into a set of smaller items with specified quantities and sizes by optimizing an objective function, such as minimizing the total waste or minimizing the cost of the cut objects. There are many studies of the CSP from different perspectives and

D. N. do Nascimento
Departamento de Engenharia de Produção, FEB, UNESP, Bauru, SP, Brazil.
E-mail: douglas.nogueira@unesp.br

S. A. de Araujo
Departamento de Matemática, IBILCE, UNESP, São José do Rio Preto, SP, Brazil.
E-mail: silvio.araujo@unesp.br

A. C. Cherri
Departamento de Matemática, FC, UNESP, Bauru SP, Brazil.
E-mail: adriana.cherri@unesp.br

their proposed solution methods in the literature, as we can see in Erjavec et al. (2012). In that paper, the authors proposed a method to define the optimal stock size for the expected demand of items to minimize the total cost of waste, warehousing and non-fulfilment. After tests and simulations, the authors concluded that the optimal stock size must be approximately 50% above the expected demand of items. Another example is the recent application studied in Lemos et al. (2020). Based on a real-life application of the concrete pole manufacturing, the authors considered multiple manufacturing modes integrated into the cutting stock problem.

A variation of CSP that often appears in practice consists of considering the generation of usable leftovers of cut objects. If planned, these leftovers are not considered waste and they are kept in stock to meet future demand. This variation of the CSP is called the Cutting Stock Problem with Usable Leftovers (CSPUL).

In production planning for multiple time periods, the possibility of generating leftovers is more complex, since this decision can be made looking not only at the demand of a single period but considering the demand of all periods of a planning horizon. In this multiperiod scenario, there is a new problem along with the CSPUL: the Lot-Sizing Problem (LSP).

The LSP determines the production size of different types of items in each period of a planning horizon, aiming to minimize an objective function, such as setup and inventory costs. The integration with the CSPUL makes it possible to bring forward the production of items or not. This bringing forward depends on the storage costs and the possible combinations of items in the objects and leftovers. Also, objects not used in a period, as well as all generated leftovers, are stored for the next period.

Integrating LSP and CSPUL, decisions about generating leftovers or bringing forward the production of items must be taken. In a situation in which a company has infinite production capacity, it could be more interesting to bring forward the production of ordered items than to delay this production by generating usable leftovers. However, in real-world situations, companies have limited item production capacity. Therefore, correctly planning which items will be brought forward and in what quantities, as well as generating leftovers with predefined sizes to produce items in the next periods, are difficult decisions that will be supported by the integration studied in this paper. For convenience, LSP-CSPUL refers to the integration of these two problems.

This paper contributes to the literature by proposing a mathematical model to represent LSP-CSPUL. The column generation method was used to solve the linear relaxation of the model and a heuristic procedure was proposed to find the integer solution. Computational tests were performed and showed the contribution of the proposed mathematical model to the literature and the quality of the solutions obtained.

The remainder of the paper is organized as follows: Section 2 is a review of the literature. In Section 3, the LSP-CSPUL is defined together with the proposed mathematical model. The solution method and the heuristic used to find integer solutions are described in Section 4. In Section 5, computational tests varying the maximum number allowed for leftovers and the inventory costs are presented. Conclusions are presented in Section 6.

2 Literature review

The CSPUL is a well-studied problem in the literature, since its practical application provides valuable benefits in various industries. Also, some papers in the literature have studied the integration of LSP and CSP but without the possibility of generating leftovers. Therefore, this section presents a literature review separating papers by CSPUL and integrated LSP-CSP.

2.1 Cutting stock problem with usable leftovers

The CSPUL was first mentioned by Brown (1971) and a considerable number of papers have been published on this problem since then. Cherri et al. (2014) presented a survey of the existing papers that investigated the one-dimensional CSPUL.

Considering one-dimensional problems, Scheithauer (1991) used the column generation technique proposed by Gilmore and Gomory (1963) to deal with the possibility of generating leftovers. In this approach, fictitious items (leftovers) were considered without a demand to be met. These items, if cut, returned to stock to meet future demand. Gradisar et al. (1997) proposed a model to represent the CSPUL with two objective functions: minimizing the number of items whose demand is not satisfied and minimizing the total waste. A heuristic procedure was also proposed and used to solve the problem. All leftovers larger than a specific threshold returned to stock. That study was applied in the clothing industry. Cherri et al. (2009) modified classic constructive and residual heuristic procedures to solve the one-dimensional CSPUL.

Abuabara and Morabito (2009) studied the one-dimensional CSPUL in a Brazilian aeronautical company. The authors proposed a mixed integer mathematical model that is an adaptation of the model proposed by Gradisar et al. (1997). Cui and Yang (2010) extended the model from Scheithauer (1991) including upper bounds on the number of leftovers, and limiting the number of objects in stock.

In Cherri et al. (2011) the authors analyzed the solution methods presented in Cherri et al. (2009) using fuzzy inference. Arenales et al. (2015) proposed a mathematical model to solve the CSPUL with the objective of minimizing the waste of material. In their model, leftovers have lengths and limited stock quantities previously defined and can be generated for stock to reduce the waste. The problem was solved using the column generation technique and optimal continuous solutions were presented.

Coelho et al. (2017) proposed a mathematical model and two heuristics to solve the one-dimensional CSPUL, considering the leftovers could be returned to stock or sold to other companies. They also discuss the sustainable implications of the usable leftovers, which include reducing the environmental impact of industries, increasing their profits, and improving company image, among others. Tomat and Gradisar (2017) analyzed the possibility of generating leftovers considering consecutive demands. The method proposed aims to find the best quantities of leftovers for the stock and the ideal length for the new leftovers generated.

Birgin et al. (2019) proposed a multiperiod framework for the two-dimensional non-guillotine cutting problem with usable leftovers, aiming to minimize the cost of the cut objects in a time horizon. In the strategy used, leftovers are obtained through horizontal and vertical guillotine precuts.

2.2 Integrated lot-sizing and cutting stock problems

Some authors have proposed the integrated LSP-CSP from different perspectives. Poltroniere et al. (2008) proposed an integrated mathematical model to represent the LSP in the production of large paper rolls and the CSP to produce smaller rolls. Two decomposition heuristics were also proposed to solve the model by approaching one problem at a time, and using the solution of the first problem as a parameter in the second problem.

Poldi and de Araujo (2016) investigated the multiperiod CSP, which can be considered as an integrated LSP-CSP since the proposed mathematical model contains decision variables related to both objects and inventory items, as well as the cutting of objects. The proposed model is based on mixed integer models from the literature. A residual heuristic procedure based on rolling horizon strategies was also proposed to solve the problem. Leao et al. (2017) studied the integration of the one-dimensional CSP with the lot-sizing problem with focus on the paper industry. The authors investigated strategies based on Dantzig–Wolfe decomposition and column generation techniques to obtain good upper and lower bounds for the problem. A rounding heuristic was proposed to find feasible solutions, as well as a method combining neighborhood search and column generation.

Campello et al. (2019) addressed the integrated LSP-CSP using a multi-objective approach to find a set of pareto-optimal solutions and assess the trade-off between both problems. The authors proposed a multi-objective model and used two methods to find solutions: i) a weighting method, which consists of associating the objective function of each problem with coefficients and minimizing the weighted sum of the objective functions; ii) a constraint method, which consists of minimizing the objective function of just one of the problems and convert the other one into a constraint, setting an upper bound for it.

More details of these and other papers can be found in Melega et al. (2018) that proposed a classification of the literature related to the integrated LSP-CSP. A deterministic mathematical model, that considers multiple dimensions of integration and comprises several aspects found in practice, was proposed. This model was used as a framework to classify the current literature for this particular problem. The classification is organized around the integration of time periods and production. Within a given planning horizon consisting of several periods, the stock gives the connection between the periods, which is the first factor of integration. The other factor of integration considers three levels of production. The first one is related to the objects and include the fixed setup (or ordering) cost for the production (or purchase) of objects, their production (or purchasing) cost and their inventory holding cost may be included in the objective function. The second level refer to the cutting process, the setup cost of a cutting pattern, the cost of cutting an object according to a cutting pattern and the cost of holding the pieces in inventory may be included in the objective function. The third one correspond to the setup, production and inventory costs o final products. Following these classification criteria, this paper deals with both types of integration (across time periods and between production levels) and is restricted to two production levels (Level 1 and Level 2) since the model has decision variables related to the production (leftover) and cutting of objects in a discrete and multi-period planning horizon. According to the proposed classification, the present paper is classified as L1/L2/-/M.

3 Problem definition

In the LSP-CSPUL a finite planning horizon is divided into T periods. In each period, S types of objects are available in stock and m types of items must be cut to attend a specified demand d_{it},$i = 1, ...,m$, $t = 1, ..., T$. As the demand is known for all periods, the production of items can be brought forward to a certain period t, at the cost of storing them until the period that they were needed. Instead of this, leftovers can be generated, being available in stock for the following periods (t+1, t+2, ..., T), also at the cost of storing. These costs represent the space occupied by the stock. The bringing forward of items can allow better combinations of items in cutting patterns, consequently decreasing the waste of material. However, due to the limited capacity of production, leftovers can be necessary and preferable to bringing forward items. There is a trade-off between generating a leftover and bringing forward the production of items. The LSP-CSPUL has the objective of minimizing the waste of material and the cost of storing objects, leftovers and items.

The mathematical model proposed to solve the LSP-CSPUL is an extension of the model presented in Arenales et al. (2015), that is an extended formulation of the problem (Gilmore and Gomory (1963)), which guarantees high quality lower bounds and directly affects the quality of the proposed solution method. The following data were used in the model:

Indexes:

- $s = 1,...,S$: number of types of standard objects;
- $k = 1,...,R$: number of types of leftovers in stock;
- $t = 1,...,T$: number of periods of time;
- $i = 1,...,m$: number of types of ordered items.

Parameters:

- d_{it}: demand for item i in period t;
- J_{st}: set of cutting patterns for object s in period t;
- $J_{st}(k)$: set of cutting patterns for object s generating a leftover k in period t;
- Jr_{kt}: set of cutting patterns for leftover k in period t;
- a_{ijst}: number of items i in cutting pattern j for object s in period t, $j \in J_{st}$;
- $a_{ijst}(k)$: number of items i in cutting pattern j for object s generating a leftover k in period t, $j \in J_{st}(k)$;

- ar_{ijkt}: number of items i in cutting pattern j for leftover k in period t, $j \in Jr_{kt}$;
- U_t: maximum number of leftovers available in stock at the end of period t;
- Cap_t: production capacity in period t;
- c_{jst}: waste of cutting object s according to cutting pattern j in period t, $j \in J_{st}$;
- $c_{jst}(k)$: waste of cutting object s according to cutting pattern j when generating a leftover k in period t, $j \in J_{st}(k)$;
- cr_{jkt}: waste of cutting leftover k according to cutting pattern j in period t, $j \in Jr_{kt}$;
- L_s: length of the object s;
- Lr_k: length of the leftover k;
- l_i: length of item i;
- py_{it}: Inventory cost of item i at the end of the period t;
- pz_{kt}: Inventory cost of leftover k at the end of the period t.

Variables:

- x_{jst}: number of objects s cut according to cutting pattern j in period t, $j \in J_{st}$;
- $x_{jst}(k)$: number of objects s cut according to cutting pattern j and generating a leftover k in period t, $j \in J_{st}(k)$;
- xr_{jkt}: number of leftovers k cut according to cutting pattern j in period t, $j \in Jr_{kt}$;
- y_{it}: number of items i brought forward to period t;
- z_{kt}: number of leftovers k generated or not used in period t and available in period $t + 1$ and in the following periods.

Mathematical model:

$$\min \sum_{t=1}^{T} \Big(\sum_{s=1}^{S} \sum_{j \in J_{st}} c_{jst} x_{jst} + \sum_{s=1}^{S} \sum_{k=1}^{R} \sum_{j \in J_{st}(k)} c_{jst}(k) x_{jst}(k) + \sum_{k=1}^{R} \sum_{j \in Jr_{kt}} cr_{jkt} xr_{jkt} + \sum_{i=1}^{m} py_{it} y_{it} + \sum_{k=1}^{R} pz_{kt} z_{kt} \Big) \tag{1}$$

Subject to:

$$\sum_{s=1}^{S}\sum_{j\in J_{st}} a_{ijst}x_{jst} + \sum_{s=1}^{S}\sum_{k=1}^{R}\sum_{j\in J_{st}(k)} a_{ijst}(k)x_{jst}(k) + \sum_{k=1}^{R}\sum_{j\in Jr_{kt}} ar_{ijkt}xr_{jkt} +$$

$$y_{i,t-1} - y_{it} = d_{it}, \forall i, t \tag{2}$$

$$\sum_{j\in Jr_{kt}} xr_{jkt} \leq z_{k,t-1}, \forall k, t \tag{3}$$

$$\sum_{s=1}^{S}\sum_{j\in J_{st}(k)} x_{jst}(k) - \sum_{j\in Jr_{kt}} xr_{jkt} + z_{k,t-1} = z_{kt}, \forall k, t \tag{4}$$

$$\sum_{s=1}^{S}\sum_{k=1}^{R}\sum_{j\in J_{st}(k)} x_{jst}(k) - \sum_{k=1}^{R}\sum_{j\in Jr_{kt}} xr_{jkt} \leq U_t - \sum_{k=1}^{R} z_{k,t-1}, \forall t \tag{5}$$

$$\sum_{i=1}^{m}(\sum_{s=1}^{S}\sum_{j\in J_{st}} a_{ijst}x_{jst} + \sum_{s=1}^{S}\sum_{k=1}^{R}\sum_{j\in J_{st}(k)} a_{ijst}(k)x_{jst}(k) +$$

$$\sum_{k=1}^{R}\sum_{j\in Jr_{kt}} ar_{ijkt}xr_{jkt}) \leq Cap_t, \forall t \tag{6}$$

$$x_{jst} \in Z^+, \forall j \in J_{st}, s, t \tag{7}$$

$$x_{jst}(k) \in Z^+, \forall j \in J_{st}(k), s, k, t \tag{8}$$

$$xr_{jkt} \in Z^+, \forall j \in Jr_{kt}, k, t \tag{9}$$

$$y_{it} \in Z^+, \forall i, t \tag{10}$$

$$z_{kt} \in Z^+, \forall k, t \tag{11}$$

In the model (1)–(11), the objective function (1) minimizes the total cost of cutting objects and leftovers in stock in all periods and the cost of storing items and new leftovers. The cost parameters associated with the cutting patterns, which must be minimized, represent the waste of material, and are calculated as follows:

- $c_{jst} = L_s$ - $\sum_{i=1}^{m} l_i a_{ijst}$
- $c_{jst}(k) = (L_s$-$Lr_k)$ - $\sum_{i=1}^{m} l_i a_{ijst}(k)$
- $cr_{jkt} = Lr_k$ - $\sum_{i=1}^{m} l_i ar_{ijkt}$

Constraint (2) ensures that the demand is met. Constraint (3) ensures that the quantity of leftovers used during the cutting process in each period does not exceed their availability. Constraint (4) guarantees that, at the end of each period t, the number of leftovers in stock available for the period t+1 is equal to the number of leftovers available in stock at the begin of period t plus the number of generated leftovers minus the number of cut leftovers. Constraints (3) and (4) together ensure that a leftover is not generated and cut in the same period. Constraint (5) limits the quantity of leftovers that can be generated during the cutting process in each period. The production capacity is related to the number of items that can be generated in each period and is limited by constraint (6). Constraints (7)–(11) are the integrality and non-negativity constraints of the variables.

In practice, the proposed model solves just a temporary instance of the problem. Much of the data used in the model, for example forecasts of future demand, almost always change as time rolls forward. The only decisions that are actually implemented are those in the time interval between the initial periods of successive optimizations. Production in later periods is only represented so that its impact on nearer and more immediate periods is taken into account. To have better decision making for these later periods, the model can be executed again, with updated data, at any point of the planning horizon.

4 Solution method

Because of the integrality conditions and the exponential number of variables, it is difficult to find the optimal solution of the model (1)–(11). Therefore, these conditions are relaxed and continuous solutions for the LSP-CSPUL can be found using the simplex method with column generation (Gilmore and Gomory, 1963), which is an efficient strategy to solve linear problems with high a number of variables. Afterwards, a heuristic is applied to find integer solutions.

The continuous solution found indicates the frequency of each cutting pattern, i.e., how many objects will be cut according to each cutting pattern. The problem of determining these frequencies is called the *master problem*. In the first iteration of the solution method, homogeneous cutting patterns are generated to find an initial solution. These homogeneous cutting patterns, which produce only one type of item, cut only objects from the stock, since it is considered that the company has objects in sufficient quantity to meet the demand.

In order to find better solutions to the master problem, the model (1)–(11) must have the best possible cutting patterns available in each period of time. These cutting patterns are generated iteratively by the column generation method by solving a knapsack problem for every possible object that can be cut. The problems solved by this strategy are called *sub-problems*. In each period, three categories of objects can be cut:

- Object s, with length L_s;
- Leftover k, with length Lr_k;
- And reduced objects resulting from object s generating leftover k, with length $(L_s - Lr_k)$.

At each iteration of the solution method, a knapsack problem is solved for all objects in the categories described above and for each period. For example, consider a problem with S=1 type of object, R=3 types of leftovers and T=5 periods. In this scenario, $(S+R+S*R)*T = 35$ and so 35 knapsack problems are solved at each iteration. The solution of the objective function of the knapsack problems is used to calculate the *reduced cost* of the new column generated.

To describe the calculation of the reduced cost, the vector (π^2_{it}, π^3_{kt}, π^4_{kt}, π^5_t, π^6_t) represents the dual variables associated with constraints (2)-(6). The variable a_i indicates the number of items i in the new column (cutting pattern).

In a period t, the reduced cost β of a new column generated for each category of object is:

- $\beta_{st} = L_s - \sum_{i=1}^{m}(l_i + \pi^2_{it} + \pi^6_t)a_i$;

- $\beta_{st}(k) = (L_s - Lr_k) - \pi^4_{kt} - \pi^5_t - \sum_{i=1}^{m}(l_i + \pi^2_{it} + \pi^6_t)a_i$;
- $\beta_{kt} = Lr_k - \pi^3_{kt} + \pi^4_{kt} + \pi^5_t - \sum_{i=1}^{m}(l_i + \pi^2_{it} + \pi^6_t)a_i$.

For the example described previously, 35 reduced costs are calculated in each iteration. The cutting patterns (columns) associated with negative reduced costs are included in the master problem. After this procedure, the linear relaxation of the master problem is solved again. This procedure is repeated until no subproblems provide a new cutting pattern with a negative reduced cost.

In real-world applications, it is impossible to consider continuous solutions for cutting stock problems. Thus, a heuristic procedure adapted from the classic *relax-and-fix heuristic* (Wolsey, 1998) was proposed and used to find integer solutions. This proposed heuristic procedure consists of:

- **Step 1** - Find an optimal solution for the model (1)–(11) with the integrality conditions relaxed using the column generation method;
- **Step 2** - From all the cutting patterns generated during Step 1, select those that meet one of the following criteria:
  - Homogeneous cutting patterns;
  - Cutting patterns that compose the base of the optimal relaxed solution;
  - For cutting patterns from objects, select those with waste equal or lower than 0.5% of the length of the object. For cutting patterns from leftovers, select those with waste equal or lower than 1% of the length of the leftover.
- **Step 3** – Separate the cutting patterns selected in Step 2 into groups, which will be used in the *relax-and-fix* strategy, starting at Step 4. The groups of variables must be defined as follows:
  - Group A: variables associated with cutting patterns that cut standard objects and generate leftovers ($x_{jst}(k)$);
  - Group B: variables associated with cutting patterns that cut leftovers (xr_{jkt});
  - Group C: variables associated with cutting patterns that cut standard objects (x_{jst}). Since there are a large number of variables in this group, it is divided into T sub-groups $C_t, t = 1, ...T$, according to the number of time periods;
  - Group D: variables associated with the bringing forward of items (y_{it}) and stock of leftovers (z_{kt}).
- **Step 4** – Solve the model (1)–(11) relaxing all the variables, except those in Group A;
- **Step 5** – Fix the variables in Group A and solve the model (1)–(11) relaxing all the variables, except those in Group B;
- **Step 6** – Fix the variables in Group B and solve the model (1)–(11) relaxing all the variables, except those in Sub-group C_1;
- **Step 7** – For each period $t' = 2...T$, fix the variables in Sub-group $C_{t'-1}$, and solve the model (1)–(11) relaxing all the variables, except those in Sub-group $C_{t'}$;
- **Step 8** – Fix the variables in Sub-group C_T and solve the model (1)–(11) considering the integrality constraints of the variables in Group D. All the variables in the solution obtained after this step will have integer values.

For some instances, due to the capacity constraint (6), an infeasibility may occur during the last iteration of Step 7, when the variables associated with cutting patterns that cut standard objects in the last period are not relaxed. In this case, the procedure must be restarted, adding the following constraint to the model:

$$\sum_{i=1}^{m}\sum_{t=1}^{T-1}(\sum_{s=1}^{S}\sum_{j\in J_{st}} a_{ijst}x_{jst} + \sum_{s=1}^{S}\sum_{k=1}^{R}\sum_{j\in J_{st}(k)} a_{ijst}(k)x_{jst}(k) + \sum_{k=1}^{R}\sum_{j\in Jr_{kt}} ar_{ijkt}xr_{jkt}) \geq \sum_{i=1}^{m}\sum_{t=1}^{T} d_{it} - Cap_T + \alpha \tag{12}$$

Constraint (12) ensures that the quantity of items produced in the first $T-1$ periods is sufficient to leave for period T only $Cap_T - \alpha$ items to be produced, with α being an integer number greater than 0. When the infeasibility occurs for the first time, the constraint is included considering $\alpha = 1$. If the infeasibility remains, update $\alpha = \alpha + 1$ and restart the procedure, until an integer solution is found.

5 Computational tests

In this section, we present the description and the results of the computational tests. These tests considered two situations: problems with just one type of standard object in stock, with sufficient availability to meet the demand; and problems with more than one type of standard object, in limited quantity in stock, and with inventory costs for these objects.

To evaluate the performance of the proposed approach, the model (1)–(11) was coded using OPL (Optimization Programming Language) with the CPLEX, version 12.10 solver. The computational tests were run on an Intel Core i7, 2.8 GHz,16 GB RAM computer.

5.1 Data sets

In order to simulate a planning horizon in a five-working-day week, the tests considered T=5 periods of time. The values of other parameters were based on papers which also consider usable leftovers in cutting stock problems (Arenales et al. (2015)) and the integration of lot-sizing and cutting stock problems (Poldi and de Araujo (2016)). Some adaptations were made with the aim of reproducing different scenarios in which both the generation of leftovers and the bringing forward of items are good alternatives.

Tests considered m=10 types of ordered items. The length of items (l_i) was randomly generated in the interval [150, 800] and the values for the demand (d_{it}) were randomly generated in the interval [100, 400].

For the first tests, there is only one type of object (S=1) in stock, with length L_s=1500. Three types of leftovers were considered with lengths (Lr_k) 650, 800 and 1000. The availabilities of all types of leftovers were 0 for the first period. The maximum quantity of leftovers in each period varied as $U_t = 0$, 1 and 5. The inventory cost for items $py_{it} = \lambda^y l_i$ varied for $\lambda^y = 0.00$, 0.01 and 0.05. In the last period an extra cost equals 0.5 was considered, because there is no information about the demand for the next planning horizon. The inventory cost of leftovers $pz_{kt} = \lambda^z Lr_k$ varied for $\lambda^z = 0.00$, 0.01 and 0.05.

Considering that there is a trade-off between generating items and leftovers, different classes of instances were tested by varying the inventory costs for items and leftovers, and the parameter U_t. Only classes in which the inventory cost for items is equal or greater than the inventory cost for leftovers were considered, since storing items requires more careful planning until they are delivered to customers. For each class, 15 instances were randomly generated. Table 1 describes the classes.

Table 1: Definition of the classes

Class	U	λ^y	λ^z
1	0	0.00	0.00
2	0	0.01	0.00
3	0	0.05	0.00
4	1	0.00	0.00
5	1	0.01	0.00
6	1	0.01	0.01
7	1	0.05	0.00
8	1	0.05	0.01
9	1	0.05	0.05
10	5	0.00	0.00
11	5	0.01	0.00
12	5	0.01	0.01
13	5	0.05	0.00
14	5	0.05	0.01
15	5	0.05	0.05

These classes were tested for two different scenarios of production capacity. In the first scenario, the production capacity in each period was equal to the maximum number of ordered items in any period. In the second scenario, the production capacity in each period was equal to 101% of the average quantity of ordered items. By comparing these two scenarios, the behavior of the proposed model with both a large and a restricted production capacity could be verified.

- **Scenario 2:** $Cap_t = 1.01(\sum_{i=1}^{m} \sum_{t=1}^{T} d_{it}/T)$.

5.2 Computational results with one type of standard object

The results are presented for two situations: solving the mathematical model relaxing the integer conditions of the variables (continuous solutions); and using the proposed heuristic to obtain integer solutions.

5.2.1 Continuous solutions

Tables 2 and 3 show the average results of the continuous solutions for all classes tested in scenarios 1 and 2 of production capacity, respectively. The average total cost (Total Cost), the average number of items brought forward (Items Brought Forward), the average quantity of generated (Gen.) and cut (Cut) leftovers and the computational time (Computational Time) in seconds are given.

Table 2: Average results for continuous solutions for Scenario 1.

Class	Total Cost	Items Brought Forward	Leftovers Gen.	Leftovers Cut	Computational Time (s)
1	84479.44	6104.77	0	0	5.43
2	86323.49	498.07	0	0	5.61
3	90621.59	225.35	0	0	5.90
4	84467.66	5801.46	1.67	0.93	24.00
5	86304.80	496.96	2.33	1.53	27.99
6	86315.58	498.09	0.73	0.13	29.63
7	90585.41	224.39	2.40	1.60	30.79
8	90601.26	224.73	1.60	0.87	32.16
9	90619.09	225.35	0.20	0	33.94
10	84420.53	5777.24	8.08	4.41	31.46
11	86230.42	491.12	11.09	7.09	35.66
12	86284.11	498.18	3.58	0.58	36.32
13	90441.03	220.66	11.56	7.56	37.08
14	90520.07	222.28	7.67	4	38.08
15	90609.09	225.35	1	0	39.16

Table 3: Average results for continuous solutions for Scenario 2.

Class	Total Cost	Items Brought Forward	Leftovers Gen.	Leftovers Cut	Computational Time (s)
1	85560.45	2304.02	0	0	5.33
2	91667.47	2000.64	0	0	5.37
3	114253.12	1967.28	0	0	5.49
4	85541.07	2326.32	1.40	0.60	26.25
5	91642.27	2000.33	1.80	1	29.64
6	91653.50	2000.60	0.67	0.13	30.42
7	114213.58	1967.38	1.72	1.05	31.90
8	114231.46	1967.22	0.99	0.43	32.78
9	114249.95	1967.28	0.27	0	33.88
10	85463.55	2321.94	8.67	4.67	32.16
11	91541.92	1999.02	8.84	4.84	34.61
12	91597.81	2000.47	3.33	0.67	36.59

13	114061.87	1966.70	9.12	5.79	36.75
14	114148.77	1966.78	4.47	1.78	38.34
15	114237.29	1967.28	1.33	0	39.62

The average total cost for all classes was lower in Scenario 1, with a larger production capacity. In this scenario, there was a substantial difference in the number of items brought forward between the classes with $\lambda^y = 0$ (classes 1, 4 and 10) and the classes with $\lambda^y > 0$. The difference occurs because, in this scenario, all periods have enough production capacity to meet their demand, so the model just brings forward items if it is profitable.

In the Scenario 2, the number of items brought forward was close for all classes, remaining around 2000 items. In this case, some periods have a demand higher than the capacity, therefore bringing forward an item is not a choice, it is necessary. In this context, there was an increase of total cost of around 6% and 26%, in comparison with Scenario 1, for the classes with $\lambda^y = 0.01$ and $\lambda^y = 0.05$, respectively, and just 1% when $\lambda^y = 0$. The results also showed that generating and cutting leftovers is a good alternative for those classes where $\lambda^y > 0$, mainly for classes 11 and 13.

The computational time for both scenarios was similar. However, it increases when leftovers can be generated and cut (classes with $U > 0$). This increase is due to the number of sub-problems that are solved at each iteration of the solution method. When U=0, there is just one type of object that can be cut (object L_s =1500) in each period, so the solution method solves just 5 sub-problems at each iteration. For the classes with $U > 0$, in each period the model can generate cutting patterns for 1 type of object, 3 types of leftovers and 3 types of reduced objects, meaning that 35 sub-problems have to be solved.

5.2.2 Integer solutions

Tables 4 and 5 show the average results of the integer solutions obtained through the proposed heuristic procedure described in Section 4 for Scenarios 1 and 2 of production capacity, respectively. All the columns of these tables are the same as in Tables 2 and 3, except for a new column that shows the gap between the integer and continuous solutions. This gap is calculated as follows:

$$Gap(\%) = \frac{100 \times (\text{Total cost for integer solution - Total cost for continuous solution})}{\text{Total cost for integer solution}}$$

Table 4: Average results for integer solutions for Scenario 1

		Items	Leftovers			Computational
Class	Total Cost	Brought Forward	Gen.	Cut	Gap (%)	Time (s)
1	86354.07	5521.87	0	0	2.17	0.63
2	88092.26	465.27	0	0	2.01	0.78
3	92011.57	204.47	0	0	1.51	2.36

4	86223.7	5035.6	1.67	0.93	2.04	0.67
5	87855.38	454.67	2.20	1.40	1.76	2.16
6	87884.34	457.47	0.73	0.13	1.79	3.92
7	92039.46	205.4	2.53	1.73	1.58	1.84
8	92096.6	203.67	1.53	0.80	1.62	26.19
9	92023.65	204.07	0.20	0	1.53	2.45
10	86213.7	5688.73	9.67	6	2.08	0.70
11	87705.02	450.4	12.07	8.07	1.68	2.69
12	87812.87	458.93	3.60	0.60	1.74	2.72
13	91827.53	201.6	13.80	9.8	1.51	25.14
14	91764.59	200.67	7.67	4	1.36	4.88
15	91928.46	203.07	1	0	1.44	4.24

Table 5: Average results for integer solutions for Scenario 2

		Items	Leftovers			Computational
Class	Total Cost	Brought Forward	Gen.	Cut	Gap (%)	Time (s)
1	87695.27	2273.87	0	0	2.43	0.79
2	94296.34	1969.8	0	0	2.79	1.26
3	116515.32	1944.07	0	0	1.94	2.76
4	87477.47	2257	1.40	0.67	2.21	0.78
5	93653.77	1967.73	1.80	1	2.15	1.11
6	93999.58	1968.27	0.67	0.13	2.50	2.59
7	116349.91	1942.87	1.87	1.20	1.84	4.67
8	116193.03	1942.27	0.87	0.33	1.69	26.46
9	116213.75	1942.07	0.33	0.07	1.69	6.23
10	87317.17	2266.53	10.67	7	2.12	1.47
11	93870.51	1966.27	10.80	6.80	2.48	2.53
12	93983.13	1968.07	3.33	0.67	2.54	3.70
13	116063.07	1942.67	9.33	6	1.72	1.82
14	115996.16	1941.87	4.47	1.80	1.59	4.79
15	116151.07	1942.07	1.67	0.33	1.65	7.52

Many characteristics of the continuous solutions remain in the results for integer solutions. Once again, the total cost in Scenario 1 was lower for all classes, as well as the behavior for the bringing forward of items and the use of leftovers. The average computational time for all classes was lower than it was for the continuous solutions, even for Classes 8 and 13, in which the average computational time was over 25 seconds. The elevated computational time for these two classes occurred due to one or two instances whose resolution lasted around five minutes. Also, the average quality of all integer solutions was highly satisfactory, with a gap lower than 3% for all classes.

5.3 Computational results with different standard objects

Some extensions, based on common practice, can be considered for the problem represented by the mathematical model (1)–(11). One possibility is to consider more than one type of standard objects in stock ($S > 1$), denominated type s with availability equal to e_{st} in period t. In this case, standard objects not used in period t will be available in period $t+1$ and the following periods. With this approach, it is necessary to add the following constraint to the model (1)–(11):

$$\sum_{j \in J_{st}} x_{jst} - w_{s,t-1} + w_{s,t} = e_{st}, \forall s, t \tag{13}$$

In constraint (13), w_{st} is a positive integer variable that indicates the number of standard objects of type s not used in period t, and available in period $t+1$. The inventory cost for each standard object of type s at the end of the period s is represented by the parameter pw_{st}. Adding this cost to (1), we have the following objective function:

$$\min \sum_{t=1}^{T} (\sum_{s=1}^{S} \sum_{j \in J_{st}} c_{jst} x_{jst} + \sum_{s=1}^{S} \sum_{k=1}^{R} \sum_{j \in J_{st}(k)} c_{jst}(k) x_{jst}(k) + \sum_{k=1}^{R} \sum_{j \in Jr_{kt}} cr_{jkt} xr_{jkt} + \sum_{s=1}^{S} pw_{st} w_{st} + \sum_{i=1}^{m} py_{it} y_{it} + \sum_{k=1}^{R} pz_{kt} z_{kt}) \tag{14}$$

To evaluate the performance of the proposed extension, computational tests were run considering the same 15 instances randomly generated for the tests in Section 5.2 (T=5 periods and m=10 types of ordered items) but with $S = 2$ standard objects in stock, with lengths $L_s = 1500$ and 2000. The availability of both types of standard objects in each period t were determined in the interval $[\lceil E_t \rceil \; \lceil 2E_t \rceil]$ (Poldi and de Araujo (2016)), where:

$$E_t = \frac{\sum_{i=1}^{m} l_i d_{it}}{\sum_{s=1}^{S} L_s}$$

We also considered the same $R = 3$ types of leftovers with lengths (Lr_k) 650, 800 and 1000, and the maximum quantity of leftovers in each period varied as U_t = 0, 3 and 5. The availability of each type of leftover in the first period was 0 when $U_t = 0$ and 1 when $U_t > 0$.

Another 15 classes of tests were considered by varying the parameter U_t, the inventory cost of items ($py_{it} = \lambda^y l_i$, $\lambda^y = 0.00$, 0.01 and 0.05) and the inventory cost of leftovers ($pz_{kt} = \lambda^z Lr_k$, $\lambda^z = 0.00$, 0.01 and 0.05). As in the previous tests, an extra cost equal to 0.5 was considered for the inventory cost of items in the last period. For all these classes, the inventory cost of standard objects was $pw_{st} = \lambda^w L_s$, $\lambda^w = \lambda^z/10$. Table 6 describes the classes.

Table 6: Definition of the classes with inventory cost for standard objects.

Class	U	λ^y	λ^z	λ^w
1’	0	0.00	0.00	0.00
2’	0	0.01	0.00	0.00
3’	0	0.05	0.00	0.00
4’	3	0.00	0.00	0.00
5’	3	0.01	0.00	0.00
6’	3	0.01	0.01	0.001
7’	3	0.05	0.00	0.00
8’	3	0.05	0.01	0.001
9’	3	0.05	0.05	0.005
10’	5	0.00	0.00	0.00
11’	5	0.01	0.00	0.00
12’	5	0.01	0.01	0.001
13’	5	0.05	0.00	0.00
14’	5	0.05	0.01	0.001
15’	5	0.05	0.05	0.005

These classes were only tested for the Scenario 2 of production capacity, since it is a more restricted and realistic situation. Tables 7 and 8 show the average results of the continuous and integer solutions, respectively, for all tested classes considering different standard objects and inventory cost for these objects.

Table 7: Average results for continuous solutions.

		Items	Leftovers		Computational
Class	Total Cost	Brought Forward	Gen.	Cut	Time (s)
1’	12389.67	2251.06	0	0	16.42
2’	18006.94	1898.68	0	0	17.16
3’	36944.52	1891.87	0	0	18.22
4’	12377.62	2228.79	3.60	4.07	46.24
5’	17987.51	1898.68	3.28	3.95	51.32
6’	25625.16	1898.68	0.80	2.80	51.50
7’	36917.14	1891.87	4.40	4.87	53.46
8’	44627.59	1891.87	0.93	2.80	55.54
9’	75354.92	1891.87	0	3	56.86
10’	12371.34	2226.24	6.25	5.45	45.80
11’	17975.53	1898.68	6.02	5.35	49.67
12’	25621.23	1898.68	1.33	2.8	50.72
13’	36897.61	1891.87	8.53	7.27	54.24
14’	44621.83	1891.87	1.60	2.80	54.45
15’	75354.92	1891.87	0	3	57.86

Table 8: Average results for integer solutions.

Class	Total Cost	Items Brought Forward	Leftovers Gen.	Leftovers Cut	Gap (%)	Computational Time (s)
1’	13241.90	2245.73	0	0	6.44	1.85
2’	18640.05	1906.20	0	0	3.40	106.48
3’	37588.74	1894.93	0	0	1.71	124.88
4’	13023.70	2218.07	3.07	4	4.96	2.68
5’	18654.88	1901.20	3	3.53	3.58	44.11
6’	26388.86	1904.47	1.27	2.93	2.89	30.54
7’	37510.28	1893.73	3.67	4.20	1.58	13.35
8’	45261.22	1893.27	1.73	3.27	1.40	13.63
9’	75915.39	1893.67	0.27	3.07	0.74	160.63
10’	13279.43	2209.40	6.60	6.07	6.84	18.62
11’	18630.70	1901.07	6	5.20	3.52	30.40
12’	26162.40	1902.47	1.60	2.80	2.07	32.73
13’	37396.52	1894.40	8.07	6.33	1.33	18.99
14’	45298.11	1895.20	2.27	3.13	1.49	55.76
15’	75903.62	1894	0.27	3	0.72	149.92

The total cost for all classes were much lower than the total cost in the situation with just one type of standard object, even considering the inventory cost of standard objects. This occurred because there were more types of objects that could be cut to produce items (2 standard objects, 6 reduced objects and 3 leftovers). With more types of objects there are more possibilities of combinations and it is easier for the solution method to find better solutions in terms of total cost.

Considering just the continuous solutions for the classes with same values of inventory costs but different values of U, the average total cost was always lower in the classes with higher values of U. For the integer solutions, this behaviour did not happen for all classes, as can be seen in the results for classes 4’ and 10’. In these classes, there were no inventory costs, so the only difference between them was the parameter U. The average total cost for Class 4’, with $U = 3$, was lower than the average total cost for Class 10’, with $U = 5$.

Regarding the computational time of the integer solutions, the high average for some classes was caused by one or two instances that were more difficult to solve. For example, Class 9’ had an average computational time of 160.63 seconds. However, a single instance was solved after 2168 seconds, and the average computational time of the other 14 instances was 17 seconds.

To understand the impact of each type of cost on the optimal solution, Table 9 shows the percentage of the cost of cutting all types of objects (Waste), the inventory cost of items (py), the inventory cost of standard objects (pw) and the inventory cost of leftovers (pz) in the objective function of both situations tested.

Table 9: Percentage of each cost in the objective function for integer solutions.

	One type of standard object			Different types of standard objects			
Class	Waste	*py*	*pz*	Waste	*py*	*pw*	*pz*
1/1'	98.55	1.45	0.00	98.02	1.98	0.00	0.00
2/2'	92.77	7.23	0.00	71.28	28.72	0.00	0.00
3/3'	76.05	23.95	0.00	39.34	60.66	0.00	0.00
4/4'	98.77	1.23	0.00	98.11	1.89	0.00	0.00
5/5'	92.90	7.10	0.00	71.42	28.58	0.00	0.00
6/6'	92.86	7.13	0.01	50.94	20.16	28.81	0.10
7/7'	76.11	23.89	0.00	39.53	60.47	0.00	0.00
8/8'	76.14	23.85	0.01	32.73	50.24	16.95	0.08
9/9'	76.15	23.84	0.01	19.34	30.10	50.55	0.01
10/10'	98.77	1.23	0.00	96.73	3.27	0.00	0.00
11/11'	92.96	7.04	0.00	71.68	28.32	0.00	0.00
12/12'	92.86	7.11	0.03	50.51	20.32	29.06	0.12
13/13'	76.24	23.76	0.00	39.08	60.92	0.00	0.00
14/14'	76.10	23.86	0.04	32.72	50.24	16.94	0.10
15/15'	76.04	23.89	0.07	19.33	30.10	50.56	0.02

Table 9 shows that, in the situation with just one type of standard object, the waste represented more than 98% of the total cost in all classes without inventory costs (Classes 1/1', 4/4' and 10/10'). In these classes, the percentage of py is more than 0% due to the extra cost of bringing forward items in the last period. The percentage of waste remained above 90% for Classes 2/2', 5/5', 6/6', 11/11' and 12/12', even with inventory costs equal to 0.01. However, for the classes with $\lambda^y = 0.05$, the inventory cost of items increased considerably, representing more than 23% of the total cost, whereas the waste decreased to 76%.

In the tests considering more than one type of standard objects and inventory costs for them, there was a wide variation in the percentage of waste among the classes, since the percentage of waste was much lower than the waste generated with just one type of standard object. While the percentage of waste was 98% without inventory costs, for those classes with $\lambda^y, \lambda^w = 0.05$, the percentage of inventory costs was higher than the waste. Regarding the inventory cost of leftovers, it was negligible for all classes in both situations, since they were stored in a quantity much smaller than items and standard objects.

5.4 Comparison between LSP-CSPUL and CSPUL

The mathematical model proposed in this paper represents the integrated problem LSP-CSPUL and it is an extension of the model presented in Arenales et al. (2015). Since (1)-(11) is solved by a multiperiod strategy that allows the bringing forward of the production of items, it is expected that better solutions could be found in comparison with the lot-for-lot strategy, in which each period is solved separately without bringing any production forward. To verify this premise, all the instances and classes used in the computational tests presented previously in Subsection 5.3 were solved using the approach proposed by Arenales et al. (2015), that has a solution method based on the lot-for-lot strategy. Consequently, inventory costs for items, leftovers and standard objects are not considered. Also, the objective function of the model minimizes only the waste of material and computational results are presented exclusively for continuous solutions.

Table 10 shows the average results of the continuous solutions obtained by both strategies for tests considering different standard objects. The average waste (Waste), the percentage of waste reduction that has been achieved by the multiperiod strategy (Red.(%)), the average quantity of generated (Gen.) and cut (Cut) leftovers, and the computational time (Time) in seconds are given.

The results from the LSP-CSPUL strategy are the same as shown in Table 7 but, instead of showing the average total cost, including waste and inventory costs, Table 10 only shows the average waste, since the mathematical model proposed by Arenales et al. (2015) does not consider inventory costs.

The results from Arenales et al. (2015) are shown for Classes 1', 4' and 10' only because these are the classes with no inventory costs. For the other classes the results are the same according to the value of the parameter U. The results from the LSP-CSPUL strategy for the remaining classes are presented to demonstrate that, even considering inventory costs, the waste was reduced.

Table 10: Average waste for continuous solutions using multiperiod and lot-for-lot strategies.

		Arenales et al. (2015)				LSP-CSPUL				
U	Class	Waste	Gen.	Cut	Time (s)	Waste	Red. (%)	Gen.	Cut	Time (s)
0	1'	17046.54	0	0	13.29	12389.67	27.32	0	0	16.42
	2'					12786.58	24.99	0	0	17.16
	3'					14379.9	15.64	0	0	18.22
3	4'	16961.3	4.45	6.58	49.76	12377.62	27.02	3.60	4.07	46.24
	5'					12771.07	24.70	3.28	3.95	51.32
	6'					12759.22	24.77	0.80	2.80	51.50
	7'					14355.11	15.37	4.40	4.87	53.46
	8'					14363.67	15.32	0.93	2.80	55.54
	9'					14279.72	15.81	0	3	56.86
5	10'	16902.76	10.68	6.35	49.72	12371.34	26.81	6.25	5.45	45.80
	11'					12761.61	24.50	6.02	5.35	49.67
	12'					12752.05	24.56	1.33	2.80	50.72
	13'					14339.38	15.17	8.53	7.27	54.24
	14'					14356.77	15.06	1.60	2.80	54.45
	15'					14279.72	15.52	0	3	57.86

As can be seen in Table 10, the LSP-CSPUL strategy had lower average waste for all the classes tested, even those with inventory costs. For Classes 1', 4' and 10', the reduction of waste was around 27%, even though the approach used by Arenales et al. (2015) generated more leftovers. As expected, this reduction occurs mainly due to the possibility of bringing forward the production of items in the LSP-CSPUL strategy. Regarding the computational time, both strategies had similar average results.

6 Conclusions

This paper presents a new mathematical model, based on Arenales et al. (2015) to represent the integration of the lot-sizing problem and the one dimensional cutting stock problems with usable leftovers. The proposed model aims to determine a cutting plan for a planning horizon, considering the possibility of bringing forward the production of items and also allowing the generation of usable leftovers that can be cut in future periods.

This situation is very common in a variety of industries but the decision making is complex due to all variables and parameters of the problem, such as the different costs involved. The proposed model considers three types of cost: cost of cutting an object (waste), cost of storing items and cost of storing leftovers. By relaxing the integer variables, the column generation method (Gilmore and Gomory, 1963) was used to solve the problems. An innovative heuristic procedure adapted from the classic relax-and-fix strategy was proposed to provide real-world integer solutions. The procedure was adapted to consider the specific characteristics of the model, such as the leftovers and the bringing forward of items.

Computational tests were run to evaluate the proposed strategy and then an extension of the problem, that considers more than one type of standard object in stock. The proposed model, the solution method and the heuristic procedure were very efficient in solving the problem, providing good solutions in reasonable computational time. The results showed that, for more restricted production capacity, the total cost is larger compared to less restricted capacity.

For future research, other extensions to the problem could be proposed, such as: considering the two-dimensional cutting stock problems; considering setup costs and setup times for cutting patterns.

Acknowledgement

The authors would like to thank the editor and anonymous reviewers for their careful reading and useful comments. This research was funded by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Grant numbers 421130/2018-0, 305261/2018-5, 306558/2018-1 and 406335/2018-4) and Fundação de Amparo a Pesquisa do Estado de São Paulo - FAPESP (Grant numbers 2013/07375-0, 2016/01860-1 and 2018/16600-0).

References

1. Abuabara, A., & Morabito, R. (2009). Cutting optimization of structural tubes to build agricultural light aircrafts. *Annals of Operations Research*, 149 (1), 149-165.
2. Arenales, M. N., Cherri, A. C., do Nascimento, D. N., & Vianna, A. C. G. (2015). A new mathematical model for the cutting stock/leftover problem. *Pesquisa Operacional*, 35 (3), 1-14.

3. Birgin, E. G., Romão, O. C., & Ronconi, D. P. (2019). The multiperiod two-dimensional non-guillotine cutting stock problem with usable leftovers. *International Transactions in Operational Research*, 00, 1-27.
4. Brown, A. R. (1971). *Optimum packing and depletion, the computer in space and resource usage problem.* New York: Macdonald – London & American Elsevier Inc.
5. Campello, B. S. C., Ghidini, C. T. L. S., Ayres, A. O. C. & Oliveira, W. A. (2019). A multiobjective integrated model for lot-sizing and cutting stock problems. *Journal of the Operational Research Society*, 0 (0), 1-13.
6. Cherri, A. C., Arenales, M. N., & Yanasse, H. H. (2009). The one-dimensional cutting stock problem with usable leftover – A heuristic approach. *European Journal of Operational Research*, 196 (3), 897-908.
7. Cherri, A. C., Alem, D. J., & Silva, I. N. (2011). Inferência Fuzzy para o problema de corte de estoque com sobras aproveitáveis de material. *Pesquisa Operacional*, 31, 1–22.
8. Cherri, A. C., Arenales, M. N., & Yanasse, H. H. (2014). The one-dimensional cutting stock problem with usable leftovers – A survey. *European Journal of Operational Research*, 236 (2), 395-402.
9. Coelho, K. R., Cherri, A. C., Baptista, E. C., Jabbour, C. J. C., & Soler, E. M. (2017). Sustainable operations: The cutting stock problem with usable leftovers from a sustainable perspective. *Journal of Cleaner Production*, 167, 545-552.
10. Cui Y., & Yang, Y. (2010). A heuristic for the one-dimensional cutting stock problem with usable leftover. *European Journal of Operational Research*, 204 (2), 245-250.
11. Erjavec, J., Gradisar, M., & Trkman, P. (2012). Assessment of stock size to minimize cutting stock production costs. *International Journal of Production Economics*, 135 (1), 170-176.
12. Gilmore, P. C., & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem – Part II. *Operations Research*, 11(6), 863-888.
13. Gradisar, M., Jesenko, J., & Resinovic, C. (1997). Optimization of roll cutting in clothing industry. *Computers & Operational Research*, 24(10), 945-953.
14. Leao, A. A. S., Furlan, & M. M., Toledo, F. M. B. (2017). Decomposition methods for the lot-sizing and cutting-stock problems in paper industries. *Applied Mathematical Modelling*, 48, 250–268.
15. Lemos, F. K., Cherri, A. C., & de Araujo, S. A. (2020). The cutting stock problem with multiple manufacturing modes applied to a construction industry. *International Journal of Production Research*, 1-19, ISSN 0020-7543 (print), 1366-588X (online). DOI: doi.org/10.1080/00207543.2020.1720923.
16. Melega, G. M., de Araujo, S. A., & Jans, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research*, 271 (1), 1-19.
17. Poldi, K. C., & de Araujo, S. A. (2016). Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem. *Annals of Operations Research*, 238, 497–520.
18. Poltroniere, S. C., Poldi, K. C., Toledo, F. M. B., & Arenales, M. N. (2008). A coupling cutting stock-lot-sizing problem in the paper industry. *Annals of Operations Research*, 157, 91–104.
19. Scheithauer, G. (1991). A note on handling residual length. *Optimization*, 22 (3), 461-466.
20. Tomat, L., & Gradisar, M. (2017). One-dimensional stock cutting, optimization of usable leftovers in consecutive orders. *Central European Journal of Operations Research*, 25 (2), 473-489.
21. Wolsey, L. A. (1998). *Integer Programming.* New Jersey: Wiley-Interscience Series in Discrete Mathematics and Optimization.