European Journal of Operational Research 308 (2023) 38–53

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A stochastic programming approach to the cutting stock problem with usable leftovers

Adriana Cristina Cherri[a], Luiz Henrique Cherri[b], Beatriz Brito Oliveira[c], José Fernando Oliveira[c,*], Maria Antónia Carravilla[c]

[a] *Faculdade de Ciências, Universidade Estadual Paulista, Bauru SP, Brasil*
[b] *Newfoundland Capital Management, São Paulo SP, Brasil*
[c] *INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal*

ARTICLE INFO

Article history:
Received 18 November 2021
Accepted 8 November 2022
Available online 17 November 2022

Keywords:
Cutting
Usable leftovers
Stochastic programming
Column generation

ABSTRACT

In cutting processes, one of the strategies to reduce raw material waste is to generate leftovers that are large enough to return to stock for future use. The length of these leftovers is important since waste is expected to be minimal when cutting these objects in the future. However, in several situations, future demand is unknown and evaluating the best length for the leftovers is challenging. Furthermore, it may not be economically feasible to manage a stock of leftovers with multiple lengths that may not result in minimal waste when cut. In this paper, we approached the cutting stock problem with the possibility of generating leftovers as a two-stage stochastic program with recourse. We approximated the demand levels for the different items by employing a finite set of scenarios. Also, we modeled different decisions made before and after uncertainties were revealed. We proposed a mathematical model to represent this problem and developed a column generation approach to solve it. We ran computational experiments with randomly generated instances, considering a representative set of scenarios with a varying probability distribution. The results validated the efficiency of the proposed approach and allowed us to derive insights on the value of modeling and tackling uncertainty in this problem. Overall, the results showed that the cutting stock problem with usable leftovers benefits from a modeling approach based on sequential decision-making points and from explicitly considering uncertainty in the model and the solution method.

1. Introduction

The cutting stock problem (CSP) is a classic combinatorial optimization problem with several industrial applications. The CSP is both economically important and difficult to solve in real-world contexts, which makes it quite relevant. In the CSP, a set of demanded items must be produced by cutting objects in stock while optimizing an objective function. Consequently, there is a growing motivation to develop efficient methods that can provide good solutions to the CSP, especially in the presence of specific characteristics, constraints, and objectives that prevent the direct application of existing models and algorithms.

The possibility to generate leftovers during the cutting process is a variation of the CSP. These leftovers return to the stock as objects and can be used later to meet demand. In the literature, this problem is known as the cutting stock problem with usable leftovers (CSPUL). In the CSPUL, a set of demanded items must be produced by cutting standard objects or leftovers in stock. Similarly to the CSP, the demand and the objects in stock are known. The demand must be met while minimizing waste or solving another convenient objective function. Leftovers in limited quantities and sizes can be generated and moved to the stock and are not considered waste.

The CSPUL is very important in industrial processes because it involves generating leftovers that can be used to meet future demand from material that would otherwise be discarded, thus reducing waste. In practice, the CSPUL can be found in several industries, like furniture, but also when cutting structural tubes to produce agricultural aircraft, steel coils, plates of glass, plates for the production of metal frames, and so on.

* Corresponding author.
E-mail addresses: adriana.cherri@unesp.br (A.C. Cherri), luiz.cherri@gmail.com (L.H. Cherri), beatriz.oliveira@fe.up.pt (B.B. Oliveira), jfo@fe.up.pt (J.F. Oliveira), mac@fe.up.pt (M.A. Carravilla).

https://doi.org/10.1016/j.ejor.2022.11.013
0377-2217/

To solve the one-dimensional CSPUL, Scheithauer (1991) modified the problem proposed by Gilmore & Gomory (1963) by adding extra items (possible leftovers) but with no fixed demand. The problem had one type of object in stock and several leftovers could be generated. With the objective of minimizing waste or concentrating it into one object, Gradišar, Jesenko, & Resinovič (1997) presented a heuristic procedure to optimize the cutting of rolls in the textile industry. Trkman & Gradisar (2007) emphasized the importance of adequately adapting methods that consider the usable leftovers so that there is no accumulation of leftovers in stock and proposed a solution method that considers this condition. Abuabara & Morabito (2009) re-wrote the mathematical model proposed by Gradišar et al. (1997) and solved the CSPUL in a company that produces agricultural aircraft.

Cherri, Arenales, & Yanasse (2009) modified heuristics from the literature so that the leftovers in each cutting pattern were small enough to be discarded as waste or large enough to be added to the stock. Cui & Yang (2010) proposed an extension of the model presented in Scheithauer (1991). The objects in stock were diversified and the number of leftovers generated during the cutting process was limited. Cherri, Arenales, & Yanasse (2013) added the possibility of prioritizing the cutting of leftovers in stock to the heuristic procedures proposed in Cherri et al. (2009). To verify the use of the leftovers during the cutting process, the authors run the model along several time periods.

Cherri, Arenales, Yanasse, Poldi, & Vianna (2014) presented a review of the literature for the one-dimensional CSPUL. Arenales, Cherri, Nascimento, & Vianna (2015) proposed a mathematical model to represent the CSPUL, in which the generated leftovers had previously defined lengths and quantities. The computational tests showed a good performance from the model and a strong relationship between the number of leftovers generated and the waste reduction. Since an excessive amount of leftovers in stock is not desirable, Tomat & Gradišar (2017) proposed a method to control the stock of leftovers and to maintain it at a suitable level.

Coelho, Cherri, Baptista, Jabbour, & Soler (2017) proposed a mathematical model and heuristics procedures to solve the CSPUL. In this study, the leftovers could be returned to the stock or sold to companies. The authors discussed the sustainability implications of generating usable leftovers. do Nascimento, de Araujo, & Cherri (2020) proposed a mathematical model to represent the integration of the lot-sizing problem with the CSPUL. The objective was to minimize the cost of cutting items by allowing the production of items that have known demands in a future planning horizon to be brought forward. The authors also proposed a heuristic procedure based on a relax-and-fix strategy. Ravelo, Meneses, & Santos (2020) proposed a heuristic algorithm and two meta-heuristic approaches to solve practical and randomly generated instances from the CSPUL literature. According to the authors, the results obtained in the computational experiments were quite good for all the tested instances.

Khan et al. (2020) presented a study for an industrial problem of production planning and cutting optimization of reels at a paper mill. The problem considered the possibility of generating leftovers and was formulated as a linear programming model. The authors used the Simplex algorithm to find the solutions, which were rounded in a post-optimization procedure to satisfy integer constraints. Considering the pipes used in civil construction, Melhem, Maher, & Sundermeier (2021) proposed a mathematical model to represent the problem and also a heuristic approach to guarantee that the leftovers from different construction projects could be reused. The authors applied a genetic-based decision support system to validate the feasibility of the solution.

Although the length and quantity of leftovers have received considerable attention in works tackling the CSPUL, the demand of items is also a parameter that influences the quality of the solutions. In the published works, this parameter was always known in advance. Nevertheless, there are situations where the values of demand in subsequent periods are not known in advance. Instead, only the corresponding probability distribution is known.

Few research papers have examined randomness in the CSP. Sculli (1981) proposed an exact solution approach for the stochastic one-dimensional CSP. In this problem, the dimension of the object to be cut is uncertain due to possible defects caused by the winding process. The solution strategy considered that the parts of the object damaged at the two extremes of the object were normal random variables that were independently distributed.

Beraldi, Bruni, & Conforti (2009) considered uncertainties in customer demand and proposed a two-stage stochastic formulation to represent the CSP. The authors solved this problem using a two-phase-based algorithm. The first phase consisted of exploring the structure of the problem and decomposing it into small sub-problems. The second phase imposed coordination among the solutions to the different sub-problems. Alem, Munari, Arenales, & Ferreira (2010) presented a study of the CSP where the demand was a random variable. The authors proposed a two-stage stochastic non-linear model with recourse and solved it using the Simplex method with column generation.

Our work differs from previous works because we consider not only uncertainty in the demand for items, but also stock generated from leftovers during the cutting process. We propose an integer mathematical model for the CSPUL and combine a scenario-based approach with tools from two-stage stochastic programming with recourse. The solution strategy is based on the Simplex method with column generation (Gilmore & Gomory, 1963). We run computational experiments to analyze the impact of uncertainty on the problem and the importance of using approaches that explicitly consider uncertainty. The tests show that the proposed method can efficiently solve the stochastic CSPUL. We also discuss the impact that different demand profiles have on the optimal solution. Finally, we demonstrate that uncertainty plays a significant role in this problem and should be explicitly acknowledged in the formulations and solution methods.

The remainder of this paper is outlined as follows. In Section 2, we describe the problem and the mathematical model proposed to represent it. In Section 3, we present the solution method developed. In Section 4, we describe the generation process for instances and scenarios, and in Section 5, we present the computational experiments and analyze the solutions. Finally, in Section 6, we present the conclusions referring to the developed study.

2. The one-dimensional CSPUL under uncertainty model

The existence of uncertain parameters in mathematical formulations is well accepted in the scientific community. Nevertheless, average or worst-case values are often used in quantitative models instead of considering the several possible outcomes or the full set of realizations of uncertainty.

In several applications of the CSPUL, the demand for items is an uncertain parameter at the moment decisions need to be made. In these situations, forecasting methods are used to approximate the demand values.

In this paper, we consider a two-stage approach for the CSPUL. The demand for the item types is known in the first stage, and is unknown in the second stage, but is approximated by a finite set of possible scenarios with associate probabilities of occurrence.

There are S types of standard objects with length L_s and R types of leftovers with length H_r in stock, and their initial stock is respectively e_s and n_r.

Standard objects and leftovers can be completely cut into items (see Fig. 1(a) and (b)). Standard objects can also be partially cut, generating a reduced object that will be cut into items and a left-

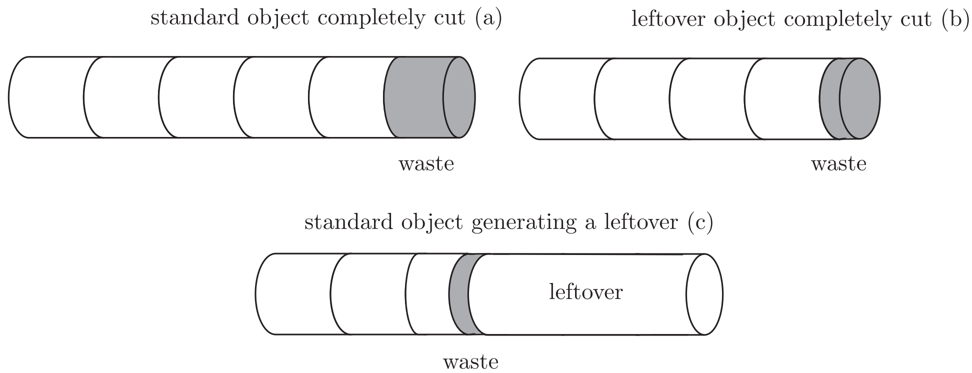


Fig. 1. Cutting patterns for standard objects and leftovers.

over with a predefined length that can be used in the second stage (see Fig. 1(c)). We consider that leftover objects do not generate leftovers.

There are costs associated with each cutting pattern for each one of the cut types represented in Fig. 1. These costs depend also on the stage. The objective is to minimize the total cost.

The decision variables of the mathematical formulation can be divided into first- and second-stage variables. First-stage decision variables are the frequency with which cutting patterns are generated for a specific deterministic problem, under the expectation of future use for the generated leftovers. These variables translate decisions that are common to all scenarios. For each combination of possible demand realizations (a scenario), there is a set of second-stage decision variables that align the deterministic and the expected problems. These decisions aim to use the available leftovers in the best possible way.

Decision variables and constraints for the first stage

For the setting in Fig. 1(a), parameters a_{ijs} represent the number of items of type i in cutting pattern j for a standard object of type s, and decision variables x^1_{js} define how often cutting pattern j is used, i.e. they define the number of objects of type s cut according to pattern j.

When a cut in a standard object of length L_s generates a leftover of length H_r, it simultaneously generates another object of length $L_s - H_r$ (see Fig. 1(c)). Parameters v_{ijsr} represent the number of items of type i in cutting pattern j for standard object of type s generating a leftover of type r, and decision variables z^1_{jsr} represent the number of objects of type s cut according to pattern j, generating a leftover of type r.

To consider the cutting of these patterns as decisions of the first stage, we use the index '1' as subscript. Expressions (1) represent the cutting of standard objects in the first stage, where A^1_s is the number of cutting patterns for standard objects and V^1_{sr} is the number of cutting patterns for standard objects generating leftovers.

$$\sum_{s=1}^{S}\sum_{j=1}^{A^1_s} a_{ijs}x^1_{js} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V^1_{sr}} v_{ijsr}z^1_{jsr}, \forall i, i = 1, \dots, m. \tag{1}$$

For leftovers in stock, parameters w_{ijr} refer to the number of items of type i in cutting pattern j for leftover of type r, and the decision variables y_{jr} represent the number of leftovers of type r cut according to pattern j.

Expressions (2) represent the cutting of leftovers in the first stage, where W^1_r is the number of cutting patterns for leftovers.

$$\sum_{r=1}^{R}\sum_{j=1}^{W^1_r} w_{ijr}y^1_{jr}, \forall i, i = 1, \dots, m. \tag{2}$$

The demand for the first stage for items of type i with length ℓ_i, d^1_i, is fulfilled by the cutting types represented in expressions (1) and (2). Constraints (3), which combine expressions (1) and (2), guarantee that the demand is fulfilled.

$$\sum_{s=1}^{S}\sum_{j=1}^{A^1_s} a_{ijs}x^1_{js} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V^1_{sr}} v_{ijsr}z^1_{jsr} + \sum_{r=1}^{R}\sum_{j=1}^{W^1_r} w_{ijr}y^1_{jr} = d^1_i,$$
$$\forall i, i = 1, \dots, m. \tag{3}$$

As in the classic CSP, restrictions are also imposed on the stock. For the CSPUL, the stock balance must be imposed for standard objects and leftovers.

Considering e_s as the initial stock of standard objects of type s, the stock balance constraints for these objects in the first stage are represented in (4).

$$\sum_{j=1}^{A^1_s} x^1_{js} + \sum_{r=1}^{R}\sum_{j=1}^{V^1_{sr}} z^1_{jsr} \le e_s, \forall s, s = 1, \dots, S. \tag{4}$$

Considering n^1_r as the initial stock of leftovers of type r, the stock balance constraints for these leftovers in the first stage are represented in (5).

$$\sum_{j=1}^{W^1_r} y^1_{jr} \le n^1_r, \forall r, r = 1, \dots, R. \tag{5}$$

The total quantity of leftovers in stock is limited by parameter B. This limit is imposed by constraint (6) where $n^1 = \sum_{r=1}^{R} n^1_r$ represents the total quantity of initial stock of leftovers.

$$\sum_{r=1}^{R}\left[\sum_{s=1}^{S}\sum_{j=1}^{V^1_{sr}} z^1_{jsr} - \sum_{j=1}^{W^1_r} y^1_{jr}\right] + n^1 \le B. \tag{6}$$

Decision variables and constraints for the second stage

For the CSPUL with demand uncertainty, expressions (3)–(6) constrain the decisions of the first stage. These decisions impact the decisions of the second stage. Furthermore, leftovers are generated in the first stage with the expectation of being used in the second stage. Since uncertain demand is assumed for the different items, it is reasonable to approximate it by employing a finite set of scenarios $\{1, 2, \dots, \Xi\}$ with an associated probability of occurrence $\{\pi_1, \pi_2, \dots, \pi_\Xi, \}$ so that $\pi_\xi > 0$ and $\sum_{\xi=1}^{\Xi} \pi_\xi = 1$. Each scenario ξ corresponds to a demand realization $d_{i\xi}$, for each item type i.

We complete the mathematical formulation for the CSPUL with demand uncertainty by defining constraints for the second stage. All parameters and variables used in this stage are similar to those already defined for the first stage, except for the addition of index ξ, which refers to the scenarios. We also insert the index '2' to refer to the second stage. For example, the decision variables for the second stage $x^2_{js\xi}$ represent the number of objects of type s cut according to pattern j in scenario ξ. The same reasoning is used in the definition of the other decision variables. In the second stage, the demand becomes an uncertain parameter defined by $d_{i\xi}$. We define the demand constraints as:

$$\sum_{s=1}^{S}\sum_{j=1}^{A_s^2} a_{ijs}x^2_{js\xi} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^2} v_{ijsr}z^2_{jsr\xi} + \sum_{r=1}^{R}\sum_{j=1}^{W_r^2} w_{ijr}y^2_{jr\xi} = d^2_{i\xi},$$
$$\forall i,\ \xi, i = 1,\dots,m, \xi = 1,\dots,\Xi. \tag{7}$$

The stock balance constraints of standard objects and leftovers depends on the stage. In the second stage, these constraints are composed of variables from both stages. For the standard objects, the constraints are represented in (8).

$$\sum_{j=1}^{A_s^2} x^2_{js\xi} + \sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^2} z^2_{jsr\xi} + \sum_{j=1}^{A_s^1} x^1_{js} + \sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} \le e_s,$$
$$\forall s, \xi, s = 1,\dots,S, \xi = 1,\dots,\Xi. \tag{8}$$

We control the stock of leftovers by adding the number of leftovers used during the cutting process and the leftovers generated by cutting standard objects (both from the first stage) to the number of leftovers cut in the second stage. These constraints are represented in (9).

$$\sum_{j=1}^{W_r^2} y^2_{jr\xi} + \sum_{j=1}^{W_r^1} y^1_{jr} - \sum_{s=1}^{S}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} \le n^1_r,$$
$$\forall r, \xi, r = 1,\dots R, \xi = 1,\dots,\Xi. \tag{9}$$

Parameter B limits the total quantity of leftovers in stock also in the second stage. This limit is imposed by constraints (10).

$$\sum_{r=1}^{R}\left[\sum_{s=1}^{S}\sum_{j=1}^{V_{sr}^2} z^2_{jsr\xi} - \sum_{j=1}^{W_r^2} y^2_{jr\xi} + \sum_{s=1}^{S}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} - \sum_{j=1}^{W_r^1} y^1_{jr}\right] + n^1 \le B,$$
$$\forall \xi, \xi = 1,\dots,\Xi. \tag{10}$$

Objective function

The objective of this problem is to minimize costs by cutting standard objects and leftovers in stock. Thus, each decision variable from the first and second stages has an associated cost. Parameter c_{js} represents the cost of using a standard object type s and cut it according to pattern j, h_{jsr} refers to the cost of using a standard object type s and cut it according to pattern j, while generating a leftover type r. We use parameter u_{jr} to represent the cost of using a leftover type r and cut it according to pattern j. We also define the parameter π_ξ as the probability of occurrence of the scenario ξ. The formulation of the objective function is given by:

$$\min \sum_{s=1}^{S}\sum_{j=1}^{A_s^1} c_{js}x^1_{js} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^1} h_{jsr}z^1_{jsr} + \sum_{r=1}^{R}\sum_{j=1}^{W_r^1} u_{jr}y^1_{jr} +$$
$$+ \sum_{\xi=1}^{\Xi}\pi_\xi\left[\sum_{s=1}^{S}\sum_{j=1}^{A_s^2} c_{js}x^2_{js\xi} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^2} h_{jsr}z^2_{jsr\xi} + \sum_{r=1}^{R}\sum_{j=1}^{W_r^2} u_{jr}y^2_{jr\xi}\right] \tag{11}$$

The expression for the costs is the following:

- $c_{js} = L_s, \quad \forall j, s$
- $h_{jsr} = (L_s - L_r) + (\mu + \gamma)L_r, \quad \forall j, s, r$
- $u_{jr} = (1-\gamma)L_r, \quad \forall j, r$

We are assuming that the costs are proportional to the amount (length) of raw-material used to satisfy the demand. While the cost of using a standard object (without generating any leftover) is just the length of the leftover, the cost of using a standard object and cutting it generating a leftover is equal to the length of the standard object minus the length of the generated leftover, plus a penalization (proportional to the leftover size) that accounts for two factors. Parameter γ accounts for the estimated waste that a leftover will generate in the future when it is actually cut into items, i.e., it is not expectable that in the future, a leftover will be cut without generating any waste. The parameter μ accounts for the cost of keeping a leftover in stock. Finally, when we use a leftover (cost u_{jr}), the waste estimation made when the leftover was generated is discounted (the leftover cost is smaller), as it will be replaced by the actual waste.

All decision variables used in the formulation are integer and non-negative.

In the following we will present the complete mathematical model for the CSPUL.

Mathematical model for the CSPUL

Indexes:

- i: item types;
- s: standard object types;
- r: leftover types;
- j: cutting patterns;
- ξ: scenarios.

Sets:

- $\{1,\dots,m\}$: set of items;
- $\{1,\dots,S\}$: set of standard objects;
- $\{1,\dots,R\}$: set of leftovers;
- $\{1,\dots,A_s\}$: set of cutting patterns for standard objects;
- $\{1,\dots,V_{sr}\}$: set of cutting patterns for standard objects generating leftovers;
- $\{1,\dots,W_r\}$: set of cutting patterns for leftovers;
- $\{1,\dots,\Xi\}$: set of scenarios.

Parameters that are not dependent on the scenarios:

- S: number of types of standard objects s;
- R: number of types of leftovers r;
- e_s: number of objects of type s in the initial stock;
- n^1_r: number of leftovers of type r in the initial stock;
- n^1: total number of leftovers in the initial stock;
- d^1_i: demand for items of type i in the first stage;
- A^1_s, A^2_s: number of cutting patterns for standard objects of type s in stage 1 and 2, respectively;
- V^1_{sr}, V^2_{sr}: number of cutting patterns for standard objects of type s generating a leftover of type r in stage 1 and 2, respectively;
- W^1_r, W^2_r: number of cutting patterns for leftovers of type r in stage 1 and 2, respectively;
- a_{ijs}: number of items of type i in cutting pattern j for standard objects of type s;
- v_{ijsr}: number of items of type i in cutting pattern j for standard objects of type s generating a leftover of type r;
- w_{ijr}: number of items of type i in cutting pattern j for leftovers of type r;
- c_{js}: cost of cutting a standard object of type s according to pattern j;
- h_{jsr}: cost of cutting a standard object of type s according to pattern j generating a leftover of type r;
- u_{jr}: cost of cutting a leftover of type r according to pattern j;
- B: maximum number of leftovers that can be in stock at any time.

Uncertainty-related parameters - dependent on the scenario:

- $d^2_{i\xi}$: demand for items of type i in the scenario ξ in the second stage;
- π_ξ: probability of occurrence of the scenario ξ.

First-stage decision variables:

- x^1_{js}: number of standard objects of type s cut according to pattern j;
- z^1_{jsr}: number of standard objects of type s cut according to pattern j, generating a leftover of type r;
- y^1_{jr}: number of leftovers of type r cut according to pattern j.

Second-stage decision variables:

- $x^2_{js\xi}$: number of standard objects of type s cut according to pattern j in scenario ξ;
- $z^2_{jsr\xi}$: number of standard objects of type s cut according to pattern j, generating a leftover of type r in scenario ξ;
- $y^2_{jr\xi}$: number of leftovers of type r cut according to pattern j in scenario ξ.

Objective function:

$$\min \sum_{s=1}^{S}\sum_{j=1}^{A_s^1} c_{js}x^1_{js} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^1} h_{jsr}z^1_{jsr} + \sum_{r=1}^{R}\sum_{j=1}^{W_r^1} u_{jr}y^1_{jr} + \\ + \sum_{\xi=1}^{\Xi}\pi_\xi\left[\sum_{s=1}^{S}\sum_{j=1}^{A_s^2} c_{js}x^2_{js\xi} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^2} h_{jsr}z^2_{jsr\xi} + \sum_{r=1}^{R}\sum_{j=1}^{W_r^2} u_{jr}y^2_{jr\xi}\right] \tag{12}$$

Constraints:

$$\sum_{s=1}^{S}\sum_{j=1}^{A_s^1} a_{ijs}x^1_{js} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^1} v_{ijsr}z^1_{jsr} + \sum_{r=1}^{R}\sum_{j=1}^{W_r^1} w_{ijr}y^1_{jr} = d^1_i, \quad \forall i, \tag{13}$$

$$\sum_{j=1}^{A_s^1} x^1_{js} + \sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} \le e_s, \quad \forall s, \tag{14}$$

$$\sum_{j=1}^{W_r^1} y^1_{jr} \le n^1_r, \quad \forall r, \tag{15}$$

$$\sum_{r=1}^{R}\left[\sum_{s=1}^{S}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} - \sum_{j=1}^{W_r^1} y^1_{jr}\right] + n^1 \le B \tag{16}$$

$$\sum_{s=1}^{S}\sum_{j=1}^{A_s^2} a_{ijs}x^2_{js\xi} + \sum_{s=1}^{S}\sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^2} v_{ijsr}z^2_{jsr\xi} + \sum_{r=1}^{R}\sum_{j=1}^{W_r^2} w_{ijr}y^2_{jr\xi} = d^2_{i\xi}, \quad \forall i,\ \xi, \tag{17}$$

$$\sum_{j=1}^{A_s^2} x^2_{js\xi} + \sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^2} z^2_{jsr\xi} + \sum_{j=1}^{A_s^1} x^1_{js} + \sum_{r=1}^{R}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} \le e_s, \quad \forall s, \xi, \tag{18}$$

$$\sum_{j=1}^{W_r^2} y^2_{jr\xi} + \sum_{j=1}^{W_r^1} y^1_{jr} - \sum_{s=1}^{S}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} \le n^1_r, \quad \forall r, \xi, \tag{19}$$

$$\sum_{r=1}^{R}\left[\sum_{s=1}^{S}\sum_{j=1}^{V_{sr}^2} z^2_{jsr\xi} - \sum_{j=1}^{W_r^2} y^2_{jr\xi} + \sum_{s=1}^{S}\sum_{j=1}^{V_{sr}^1} z^1_{jsr} - \sum_{j=1}^{W_r^1} y^1_{jr}\right] + n^1 \le B, \quad \forall\ \xi, \tag{20}$$

$$x^1_{js} \in Z^+, \forall j, s;\ z^1_{jsr} \in Z^+, \forall j, s, r;\ y^1_{jr} \in Z^+, \forall j, r, \tag{21}$$

$$x^2_{js\xi} \in Z^+, \forall j, s,\ \xi;\ z^2_{jsr\xi} \in Z^+, \forall j, s, r,\ \xi;\ y^2_{jr\xi} \in Z^+, \quad \forall j, r, \xi. \tag{22}$$

In the model proposed (12)–(22), the objective function (12) minimizes the cost of cutting standard objects and leftovers in stock and the expected costs of second-stage decisions, which are also costs of cutting standard objects and leftovers. Constraints (13) and (17) refer to the fulfillment of demand in the first and second stages, respectively. Constraints (14) and (18) limit the cutting of standard objects of type s to the number of standard objects of type s in stock at each stage. Constraints (15) and (19) are equivalent to the previous ones, but for leftover objects. Constraints (16) and (20) limit to B the total number of leftovers in stock in each stage. Finally, (21)-(22) define the domains of the variables.

As already known, the integrality conditions and the exponential number of variables make it hard to solve the model (12)–(22) to optimality. A strategy often used to solve this problem consists of relaxing constraints (21)-(22) and solving the linear relaxation with the column generation method (Gilmore & Gomory, 1963) described next. In this paper, we modified this method to consider uncertainties.

3. The column generation method for the CSPUL with demand uncertainty

To solve the linear relaxation of the model (12)–(22), we initially consider that cutting pattern sets A^1_s, V^1_{sr}, W^1_r, A^2_s, V^2_{sr} and W^2_r are empty. Then, we generate a feasible solution for this formulation by adding a reduced number of columns obtained by homogeneous cutting patterns to these empty sets. These cutting patterns (columns) contain just one type of item and are very easy to obtain. It should be noticed that it is possible to use more sophisticated strategies to generate the initial set of columns, such as inserting columns generated using heuristics (for example, by disregarding the future periods), which may help the convergence of the column generation process. Without the complete set of columns, this problem is called a restricted master problem (RMP). During the column generation method, columns are periodically included in these sets.

We solve the linear relaxation of the RMP and associate the vector of dual variables (the Simplex multiplier vector), given by $\{\lambda^{13}, \lambda^{14}, \lambda^{15}, \lambda^{16}, \lambda^{17}, \lambda^{18}, \lambda^{19}, \lambda^{20}\}$, with constraints (13)–(20), respectively. We then use these variables to build the objective function of the sub-problems. Since partial cuts can be performed for each standard object type, that is, there are several types of cutting patterns, we must solve several sub-problems in each iteration of the column generation method. When all sub-problems are solved, we obtain attractive cutting patterns (columns). We include the most attractive cutting pattern in the RMP and generate a new relaxed problem. We solve the problem again and obtain new dual variables associated with the solution to the problem. We repeat the process while it generates attractive cutting patterns; otherwise, the optimal solution to the relaxed RMP is found. Algorithm 1 illustrates the column generation method used to solve the relaxed model.

According to formulation (12)–(22), six different types of columns are evaluated in each iteration of the method and the pricing information must be properly used for each column type. The new columns for the relaxed RMP solution are obtained for the first and second stages using a standard object, a standard object generating a leftover, and a leftover in stock. The probability of occurrence of the scenarios is also considered in this process.

To construct the sub-problem relating to each type of object, we must define: 1) the cost of the cutting pattern (C); 2) the length of the object used to perform the cut (H); and 3) the pricing value used to calculate the reduced costs (λ). The information needed to

Algorithm 1: column generation.

Input: Number of standard object types S, number of leftover types R, and number of item types m.
Output: Solution lower bound LB, column sets $\{A_s^1;\ V_{sr}^1;\ W_r^1;A_s^2;\ V_{sr}^2;\ W_r^2\}$.

$LB=\infty;A_s^1=\emptyset;V_{sr}^1=\emptyset;W_r^1=\emptyset;A_s^2=\emptyset;V_{sr}^2=\emptyset;W_r^2=\emptyset.$
$s=1,...,S;\ r=1,...,R;$
for $i=1,...,m$ **do**
  $a_{si}^1(i)=1;\quad a_{sz}^1(z)=0,\forall z=1,...,m,z\neq i$
  $a_{si}^2(i)=1;\quad a_{sz}^2(z)=0,\forall z=1,...,m,z\neq i$
end
$stop=false;\ j=0;$
while *stop = false* **do**
  Solve the linear relaxation of the *RMP* using the columns in $\{A_s^1;\ V_{sr}^1;\ W_r^1;A_s^2;\ V_{sr}^2;\ W_r^2\}$ and update *LB*.
  Get dual cost vectors $\lambda^{13},\lambda^{14},\lambda^{15},\lambda^{16},\lambda^{17},\lambda^{18},\lambda^{19},\lambda^{20}$ from the *RMP* solution;
  Solve the sub-problems using the corresponding dual cost vector value. Let $\{a_{sj}^{1*},v_{srj}^{1*},w_{rj}^{1*},a_{sj}^{2*},v_{srj}^{2*},w_{rj}^{2*}\}$be the best solution to each sub-problem.
  if *column* $\{a_{sj}^{1*}\}$ *is the most attractive* **then** $A_s^1=A_s^1\bigcup\{a_{sj}^{1*}\}$;
  if *column* $\{v_{srj}^{1*}\}$ *is the most attractive* **then** $V_{sr}^1=V_{sr}^1\bigcup\{v_{srj}^{1*}\}$;
  if *column* $\{w_{rj}^{1*}\}$ *is the most attractive* **then** $W_r^1=W_r^1\bigcup\{w_{rj}^{1*}\}$;
  if *column* $\{a_{sj}^{2*}\}$ *is the most attractive* **then** $A_s^2=A_s^2\bigcup\{a_{sj}^{2*}\}$;
  if *column* $\{v_{srj}^{2*}\}$ *is the most attractive* **then** $V_{sr}^2=V_{sr}^2\bigcup\{v_{srj}^{2*}\}$;
  if *column* $\{w_{rj}^{2*}\}$ *is the most attractive* **then** $W_r^2=W_r^2\bigcup\{w_{rj}^{2*}\}$;
  if *no column is attractive* **then** $stop=true$;
  $j++$;
end
return *LB* and $\{A_s^1;\ V_{sr}^1;\ W_r^1;A_s^2;\ V_{sr}^2;\ W_r^2\}$.

Table 1
Values of cost C and length H.

Stage	Cutting pattern	C	H
First stage	Std. object	L_s	L_s
	Leftover	$(1-\gamma)L_r$	L_r
	Std. object gen. a leftover	$(L_s-L_r)+(\mu+\gamma)L_r$	L_s-L_r
Second stage	Std. object	$\pi_\xi L_s$	L_s
	Leftover	$\pi_\xi(1-\gamma)L_r$	L_r
	Std. object gen. a leftover	$\pi_\xi(L_s-L_r)+(\mu+\gamma)L_r$	L_s-L_r

build each sub-problem is given in Table 1. Recalling, γ accounts for the estimated waste that a leftover will generate in the future when it is actually cut into items. If a leftover is used, this waste estimation is discounted and replaced with the actual waste. The parameter μ is the estimated cost of keeping a leftover in stock. In this table, “std” is the abbreviation of standard.

To know if the newly generated column can improve the RMP value, we need to calculate its reduced cost. We will build the expression of the reduced cost for a column of the type “standard object in the first stage”, and similar reasoning allows obtaining the same information for the other five types of columns.

For a standard object of type s with length L_s, the cost c'_{js} is the length of the object expressed by equality (23).

$$C=c'_{js}=L_s. \tag{23}$$

To improve the quality of the relaxed RMP solution, we know that the reduced cost has to be strictly negative. That is, for a pattern j for a standard object of type s the inequality (24) has to hold.

$$L_s-\sum_{i=1}^{m}\lambda_i^{13}a'_{ijs}-\lambda_s^{14}-\sum_{\xi=1}^{\Xi}\lambda_{s\xi}^{18}<0$$
$$\Rightarrow -\sum_{i=1}^{m}\lambda_i^{13}a'_{ijs}<\lambda_s^{14}-L_s+\sum_{\xi=1}^{\Xi}\lambda_{s\xi}^{18}. \tag{24}$$

The right-hand side of inequality (24) is a constant. Thus, a cutting pattern that minimizes the left-hand side is desirable.

The value of a'_{ijs} defines the frequency with which the item of type i appears in the pattern j cut using an object of type s and is equal to δ_i. Therefore, minimizing the left-hand side of expression (24) is equivalent to maximizing $\sum_{i=1}^{m}\lambda_i^{13}\delta_i$. Given the relaxed RMP solution, the most profitable cutting pattern for a standard object in the first stage can be found by the model (25)–(27).

$$\max\quad \sum_{i=1}^{m}\lambda_i^{13}\delta_i \tag{25}$$

$$\text{s.t.}\quad \sum_{i=1}^{m}\ell_i\delta_i\leq L_s,\quad s=1,\dots,S \tag{26}$$

$$\delta_i\in\mathbb{Z}_+,\quad i=1,\dots,m. \tag{27}$$

If the optimal solution δ^* for the model (25)–(27) satisfies inequality (24), then the cutting pattern obtained improves the value of the linear relaxation of the RMP. The value of this cutting pattern in the RMP is given by equality (23).

Through similar reasoning, for the first-stage columns, the sub-problem is given by:

$$\max\quad \sum_{i=1}^{m}\lambda_i^{13}\delta_i \tag{28}$$

$$\text{s.t.}\quad \sum_{i=1}^{m}\ell_i\delta_i\leq H \tag{29}$$

$$\delta_i\in\mathbb{Z}_+,\quad i=1,\dots,m. \tag{30}$$

where H is given in Table 1 and δ_i is a variable that will find the frequency with which each item is cut from a column, i.e., the value of $a'_{ijs},w'_{ijr},v'_{ijsr},t'_{ijrr'}$.

The second-stage column is generated after replacing the objective function (28) with:

$$\max\quad \textstyle\sum_{i=1}^{m}\lambda_i^{17}\delta_i \tag{31}$$

and in this case, δ_i is a variable that will find the frequency with which each item is cut from a column in the second stage, i.e., the value of $a'_{ijs\xi},w'_{ijr\xi},v'_{ijsr\xi},t'_{ijrr'\xi}$.

The correspondent pricing information, for each one of the six types of columns, is presented in Table 2.

The sub-problems presented above were solved with a MIP solver. However, a dynamic programming algorithm would also be a good choice, as it is known for being faster than MIP solvers and allows one to solve the problems related to all different leftover sizes in one single run.

4. Instance and scenario generation

In this section, we describe the process of generating the instances and scenarios analyzed. We also demonstrate the process of obtaining realistic and representative values for certain and uncertain parameters. This allows us to derive pertinent insights from

Table 2
The pricing information.

Stage	Cutting pattern	Pricing
First stage	Std. object	$C-\sum_{i=1}^{m}\lambda_i^{13}a'_{ijs}-\lambda_s^{14}-\sum_{\xi=1}^{\Xi}\lambda_{s\xi}^{18}$
	Std. object gen. a leftover	$C-\sum_{i=1}^{m}\lambda_i^{13}v'_{ijsr}-\lambda_s^{14}-\lambda^{16}-\sum_{\xi=1}^{\Xi}\lambda_{s\xi}^{18}+\sum_{\xi=1}^{\Xi}\lambda_{r\xi}^{19}-\sum_{\xi=1}^{\Xi}\lambda_{\xi}^{20}$
	Leftover	$C-\sum_{i=1}^{m}\lambda_i^{13}w'_{ijr}-\lambda_r^{15}+\lambda^{16}-\sum_{\xi=1}^{\Xi}\lambda_{r\xi}^{19}+\sum_{\xi=1}^{\Xi}\lambda_{\xi}^{20}$
Second stage	Std. object	$C-\sum_{i=1}^{m}\lambda_{i\xi}^{17}a'_{ijs\xi}-\lambda_{s\xi}^{18}$
	Std. object gen. a leftover	$C-\sum_{i=1}^{m}\lambda_{i\xi}^{17}v'_{ijsr\xi}-\lambda_{s\xi}^{18}-\lambda_{\xi}^{20}$
	Leftover	$C-\sum_{i=1}^{m}\lambda_{i\xi}^{17}w'_{ijr\xi}-\lambda_{r\xi}^{19}+\lambda_{\xi}^{20}$

the computational experiments regarding the applicability and efficiency of the solution method as well as the relevance of the proposed approach. All data is available in the online Supplementary Materials provided with this paper.

4.1. Instances

An instance is composed of the following information:

- standard objects:
  - number of types of standard objects (S);
  - length of each standard object of type s (L_s);
  - number of standard objects of type s initially in stock (e_s);
- leftovers:
  - number of types of leftovers (R);
  - length of leftovers of type r (L_r);
  - number of leftovers of type r initially in stock (n_r^1);
  - maximum number of leftovers that can be kept in stock at each moment (B).
- items:
  - number of types of items (m);
  - length of each item of type i (l_i);
  - demand for items of type i in the first stage (d_i^1);

Additionally, the overall solution approach involves the following parameters:

- estimated percentage of waste that the leftovers generated in the first stage will eventually produce in the second stage when cut into actual items (γ);
- cost of keeping one leftover in stock (μ);
- number of scenarios (Ξ).

Finally, we must define the information regarding the second stage of decision-making, including the value of the uncertain parameters in each scenario and the probability distribution underlying the scenarios:

- demand for items of type i in scenario ξ of the second stage ($d_{i\xi}^2$);
- probability of occurrence of the scenario ξ (π_ξ).

To run the computational experiments, 50 instances were randomly generated using the problem generator proposed by Arenales et al. (2015), as follows.

Standard objects

For each instance, we considered one type of standard object with length $L_1 = 1000$ and enough availability to meet the demand in any scenario.

Leftovers

Leftovers can be of 45 different lengths, starting at 420 and ending at 860, with increments of 10. The smallest leftover length (420) corresponds approximately to the average item length (421.64), rounded down to the closest dozen, and the largest leftover length (860) is equal to the length of the standard object ($L_1 = 1000$) minus the length of the smallest item ($1000 - 140 = 860$). We chose to use standardized lengths in increments of 10 since companies need to be able to easily identify the leftover in stock to use, which would be difficult if every leftover length was allowed.

Table 3
Parameters used in the computational experiments.

	Parameters					
	S	R	m	B	μ	Ξ
Values	1	45	15	2500	0.01	5

We set the maximum number of leftovers that can be kept in stock in each stage to 2500. Table A2 of the Supplementary Materials presents the initial stock of leftovers of each type for all instances. We determined these values by solving, for each instance, the deterministic first-stage problem without an initial stock of leftovers. Then, we considered the generated leftovers to be the initial stock of the instance in the following computational experiments of the stochastic programming model.

Items

For each instance, we considered $m = 15$ types of items. We randomly generated the lengths of each item type in the interval $[v_1 L_1, v_2 L_1]$, with $v_1 = 0.14$ and $v_2 = 0.7$, thus encompassing small, medium, and large items (Arenales et al., 2015). In Table A1 of the Supplementary Materials, we present in detail, for the 50 instances, the 15 item lengths generated for the computational experiments, sorted in non-decreasing order.

Solution approach parameters

The value of parameter γ is instance-dependent. As explained in Section 2, γ accounts for the estimated waste that a leftover will generate in the future when it is actually cut into items. Therefore, we compute it as follows: a classic CSP is solved with the data regarding the first stage and without leftover generation. The percentage of waste generated in the solution to this problem is used as the γ parameter value to solve this instance under the stochastic programming framework. We present the γ value of each instance in Table A1 of the Supplementary Materials.

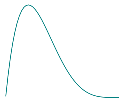
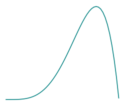
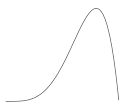
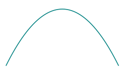
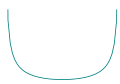
We set the value of parameter μ to 0.01 and consider 5 scenarios in the second stage of the problem.

In Table 3, we present a summary of the main non-instance-dependent parameters used in the computational experiments.

4.2. Scenarios

The scenarios are possible realizations of the uncertain parameters of the problem. Therefore, a scenario consists of concrete values of demand for each item type i, denoted by $d_{i\epsilon}^2$. To generate a representative scenario set, we developed a Monte Carlo simulation approach associated with different probability distributions. Given a probability distribution and a list of items ordered by size, the Monte Carlo simulation returns the fraction of overall

Table 4
Scenarios.

Scenario (ξ)	Demand profile		Beta(α, β)	Scenario characteristics
1		Right-tailed	(2,5)	Higher demand of smaller items
2		Left-tailed	(5,2)	Higher demand of larger items
3		Uniform	(1,1)	Similar demand levels for different-sized items
4		"Normal"	(2,2)	Symmetric profile, with higher demand for items close to the mode value of the size
5		Bathtub	(0.5,0.5)	Symmetric profile, with higher demand for the more extreme sizes (smallest and largest)

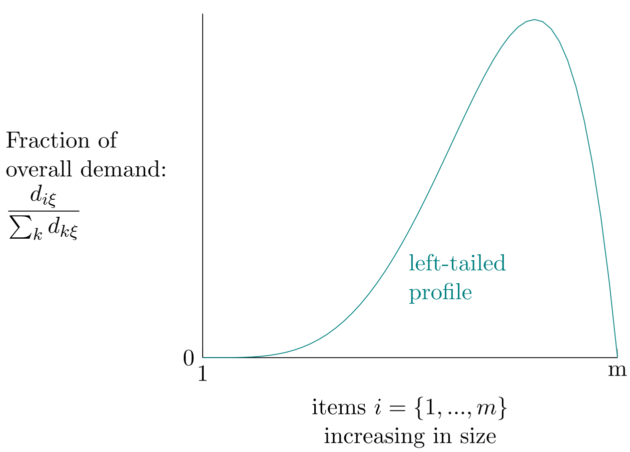


Fig. 2. Example of an asymmetric demand profile, where larger items are more requested than smaller items.

Table 5
Demand values for the different item types in the five generated scenarios ($d^2_{i\xi}$).

	Scenarios				
Item	$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$
1	109	0	133	19	342
2	258	1	143	72	141
3	323	0	139	105	113
4	320	11	141	165	107
5	271	22	142	159	75
6	226	48	134	190	94
7	176	69	121	187	89
8	125	121	135	197	89
9	91	174	149	208	80
10	63	258	125	191	91
11	25	279	133	173	83
12	9	320	121	131	91
13	2	313	128	97	121
14	2	271	123	73	133
15	0	113	133	33	351
Total demand	**2000**	**2000**	**2000**	**2000**	**2000**

demand that refers to each item. For the same list of items (i.e., for the same instance), different probability distributions design different demand profiles, resulting in scenarios that represent different practical situations. An example of a relevant demand profile is one where larger items have higher demand levels than smaller items (Fig. 2).

We considered five scenarios, or demand profiles, whose characteristics are summarized in Table 4. All profiles were obtained using a beta distribution due to its flexibility and support in [0,1]. We set the overall demand value to 2000. We present the resulting demand values in Table 5 and in Table B1 in the Supplementary Materials.

We tested three different probability distributions to describe the relative frequency of each of the five scenarios previously described. In the base case (BC), the probability of occurrence of each scenario is $\pi_\xi = \frac{1}{\Xi}$, i.e. the scenarios are equally probable. Additionally, we tested a probability distribution where scenarios with more "extreme" demand (small items and large items) are more frequent, and the opposite case, where scenarios with more average items occur more often. Table 6 summarizes the values of π_ξ tested (also in Table B2 in the Supplementary Materials),

Table 6
Cases analyzed of the probability of occurrence of each scenario (π_ξ).

	Scenarios				
Probability distribution	$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$
Base case (BC)	20%	20%	20%	20%	20%
Demand on center (DC)	10%	10%	20%	50%	10%
Demand on extremes (DE)	25%	25%	10%	5%	35%

where the scenario numbers correspond to the notation presented in Tables 4 and 5.

5. Computational results and discussion

In this section, we present and discuss the results of the computational experiments, including the measures of the uncertainty effects. The computational experiments were run on a computer with an Intel Core i5 9300H processor, 32 GB of memory, and the Ubuntu 18.04 operating system. The algorithms were implemented in C/C++ using the IBM ILOG CPLEX 12.9 optimization library.

Table 7
Final objective value (Eq. (12)) and computational time in seconds for each probability distribution case considered: BC, DC, and DE. Instances are aggregated in groups according to their quartile referring to the initial leftover stock.

		Objective value			Time (sec)		
Instance group	# inst.	BC	DC	DE	BC	DC	DE
1	14	1,967,883.1	1,951,623.2	1,977,180.3	935.0	797.1	1,070.6
2	20	1,739,691.5	1,721,789.9	1,751,200.7	1,193.9	989.0	1,290.2
3	11	1,695,055.4	1,685,994.1	1,700,882.5	1,455.9	1,419.8	1,597.5
4	5	1,665,440.1	1,657,422.3	1,672,035.4	1,136.6	1,331.4	1,246.2
	Average	**1,786,340.1**	**1,771,831.4**	**1,795,488.4**	**1,173.3**	**1,064.3**	**1,291.9**

Table 8
Upper bound of the gap of the final objective value for each probability distribution case and instance group considered.

	Average GAP_{LR}			Minimum GAP_{LR}			Maximum GAP_{LR}		
Instance group	BC	DC	DE	BC	DC	DE	BC	DC	DE
1	0,017%	0,019%	0,017%	0,000%	0,000%	0,000%	0,056%	0,048%	0,047%
2	0,013%	0,020%	0,017%	0,000%	0,000%	0,000%	0,046%	0,056%	0,047%
3	0,020%	0,014%	0,018%	0,000%	0,000%	0,000%	0,096%	0,065%	0,096%
4	0,007%	0,005%	0,013%	0,000%	0,000%	0,000%	0,033%	0,021%	0,063%

This section has three main goals and a corresponding structure. Firstly, we aim to show that the proposed method can efficiently solve the stochastic CSPUL. Then, we use the generated scenarios to further understand the impact that different demand profiles have on the optimal solution for this problem. Finally, we aim to demonstrate that uncertainty plays a significant role in this problem and should be explicitly acknowledged in the formulations and solution methods. Transversely to the three goals, we analyze the degree of sensitivity of the problem to different probability distributions associated with the scenarios. All tests were run considering the linear relaxation of the model (12)–(22), and, therefore, the values reported are lower bounds values for the optimal integer values.

5.1. Overall results

In Table 7, we present the results of the column generation method for each probability distribution considered. The instances are aggregated according to their quartile positioning in terms of their initial leftover stock. This means that the first group summarizes the instances with the lowest total length in the initial leftover stock, calculated as the number of leftovers multiplied by their length. As previously mentioned, we determined the initial leftover stock by solving the deterministic first-stage problem without an initial stock of leftovers for each instance. The complete results, detailed for each instance, are available in Table 13 of the Appendix.

The column generation method was able to solve the problem to optimality in a reasonable time and within the same order of magnitude for all instances. When considering the different instance groups, the objective value (i.e., the cost to be minimized) decreased as the initial leftover stock increased. A high number of leftovers that can be used in the second stage (which result from the initial stock and the leftovers cut in the first stage) can have two contradictory effects. On the one hand, it is cheaper to obtain the desired items from leftovers than from standard objects, decreasing the overall cost. On the other hand, if the leftovers are not needed, they generate more waste. These results suggest that this approach favors the use of leftovers, with good results on the overall objective function.

To study the impact of using the linear relaxation (LR) of the model (12)–(22), the results of the linear relaxation were compared against the ones obtained with a restricted master heuristic (RMH), i.e., solving the model (12)–(22) with integrality constraints, but restricted to the variables generated by the column generation process used in the solution of the linear relaxation of the model. It should be noted that the integrality constraint was imposed only on the first-stage variables, as the second-stage variables do not correspond to actual decisions but an estimation of the impact of future decisions and therefore do not need to be integers. The gap computed with the objective function value (OF) of these two approaches (32) is an upper bound of the actual gap of the linear relaxation solution.

$$GAP_{LR} = \frac{OF_{RMH} - OF_{LR}}{OF_{LR}} \tag{32}$$

As it can be seen in Table 8, not only the average gap is tiny, but the maximum gap, for all groups of instances, is also very small. The minimum gap is always almost zero. These results not only validate the conclusions drawn when using the linear relaxation but also validate the use of the restricted master heuristic to provide feasible integer solutions in a real-world setting.

As for the different probability distributions considered, as expected, costs were slightly lower for DC and higher for DE when compared with the BC. A significant factor when comparing probability distributions is the weight of large items, which complicate cutting problems. We discuss this effect in more detail in Section 5.2, where we analyze the impact of each demand profile. Nevertheless, the difference observed in the results for the three probability distributions (Table 7) was of small magnitude (a 0.5% increase in costs for DE and a 0.8% decrease for DC). This suggests that, for this specific problem, this method can tackle different distributions of uncertainty with a limited effect on the final result. This may result from providing the model with a representative and differentiated set of scenarios, independently of the probability distribution associated with their expected frequency.

5.2. Scenario analysis

In Table 9, we present the results for the deterministic problem when each scenario is considered individually. In short, we refer to the wait-and-see (WS) value of the problem, where we assume perfect information about future demand and solve the problem for each scenario under this assumption. This allows us to understand the impact that different demand profiles have on this problem, even in a deterministic setting. The results suggest that demand following a left-tailed profile (i.e., with a majority of large

Table 9
Results of the deterministic problem solved for each scenario individually (wait-and-see (WS) solutions).

Scenario		Objective value	
		Average	Δ vs Uniform
1	Right-tailored curve	1,494,568.7	-14.3%
2	Left-tailored curve	2,158,344.7	23.7%
3	Uniform curve	1,744,256.2	-
4	“Normal” curve	1,758,410.0	0.8%
5	Bathtub curve	1,765,241.2	1.2%

items) leads to significantly higher costs, whereas a right-tailed profile configuration (i.e., demand with a majority of small items) leads to the lowest cost value. This is expected, as discussed, since large items often lead to cutting configurations with higher waste. Interestingly, the results show that scenarios that consider the demand for items of increasing length as uniform, “Normal”, or following a bathtub profile lead to similar optimal objective values. In the two latter scenarios, the pattern for small, average, and large items is, in fact, the opposite: the “Normal” profile scenario has a predominance of average items and few small and large items, whereas the bathtub profile scenario has mostly small and large items and few average ones. This indicates that having a balanced mix of item sizes has a similar effect on the optimal values, independently of the type of “balancing mechanism” considered. These results also support the claim that it is important to consider different uncertain scenario configurations for this problem.

5.3. Measures of uncertainty impact

In this section, we aim to quantify the impact of uncertainty on this problem and the importance of using approaches that explicitly consider uncertainty, such as the proposed stochastic formulation. Two commonly used measures are the *value of the stochastic solution* (VSS) and the *expected value of perfect information* (EVPI) (Birge & Louveaux, 2011).

The VSS measures the cost of not considering uncertainty when making the decision. In this case, it determines how much is lost if the expected values replace all random variables. A significant VSS indicates that solving a deterministic model with an average of all scenarios instead of the stochastic model is not a good approximation.

To compute the VSS, we solve a simpler version of the problem: the expected value (EV) problem. Since the random variable in the model (12)–(22) is the demand, the expected value of the demand is defined as $\bar{d}_i = \left\lfloor \sum_{\xi=1}^{\Xi} \pi_\xi d_{i\xi} \right\rfloor$. Using $\bar{d}_i$, we solve the model (12)–(22) with the sources of uncertainty removed and the problem becomes deterministic, providing a solution with the value f^{EV}. Fixing the first-stage variables in the values provided by the solution to the EV problem, we can now solve the complete problem for each scenario. The expected value of using the EV solution (EEV) is the average of the several objective function values, weighted by the probability of occurrence of the respective scenarios. We calculate the value of the stochastic solution as $VSS = EVV - f^{RP}$.

The EVPI concept measures the maximum amount that could be saved if the decision-maker had complete and accurate information about the future. When considering perfect information, we can solve the model considering the a priori demand values, thus avoiding the costs associated with what could be, for each scenario, wrong first-stage decisions. Although perfect information cannot be realistically achieved, the EVPI measures how much we could gain from improving the existing knowledge on uncertainty (i.e., the scenarios and their probability distribution). According to Alem et al. (2010), a significant EVPI indicates that randomness has an important role in the problem and should not be ignored.

Since this problem consists of minimizing costs, the evaluation of the EVPI measure combines the value of the solution obtained using the model (12)–(22) (f^{RP}) and the wait-and-see (WS) value, defined as the expected value of using the optimal solution for each scenario (i.e., assuming each scenario contains accurate and perfect information about future demand). Using the model (12)–(22) for each scenario, we obtain a solution with the value f_ξ^{WS}. We calculate the expected value of using the wait-and-see solutions by $WS = \sum_{\xi=1}^{\Xi} f_\xi^{WS} \pi_\xi$. We define the EVPI measure as $EVPI = f^{RP} - WS$.

In Table 10, we present the VSS and EVPI results for the instance groups, aggregating the instances according to the initial leftover stock, as in previous results. Despite representing relatively small percentage values, overall, these values are relevant and translate a significant economic impact due to the order of magnitude of results in this context. As expected, instances with higher initial leftover stock presented better VSS and EVPI values. The results of the stochastic model improved progressively as the initial leftover stock increased (Table 7) with the exception of the fourth quartile of instances, which may be justified by the smaller number of instances that are considered in this quartile. A similar pattern is seen on the value of using stochastic approaches and the upper bound on the gains of these approaches. When we consider different probability distributions, the results suggest that it is especially important to explicitly acknowledge uncertainty for settings where the demand is more prevalent in the extreme sizes (very small or very large). In contrast, in scenarios where the demand for average items is more frequent, the value of these approaches (compared with using average values) decreases. These results allowed us to quantify this intuition and showed that the average VSS decreased 54% when moving from the base case (BC, with equiprobable scenarios) to the demand on center (DC) distribution and increased 28% from the BC to the demand on extremes (DE) distribution. Interestingly, this relationship was not obvious in the EVPI measure, whose average was higher for the BC and where differences between probability distributions were neither as significant nor as constant in all instance quartiles.

Overall, the VSS metric is the most relevant for this study. It allows us to quantify the importance of developing methods that explicitly consider uncertainty, like the stochastic model and column generation method proposed, instead of using average values in deterministic models. These results support the claim that uncertainty plays a relevant role in the CSPUL and that the model and solution method presented can be impactful in practice.

5.4. Multi-period framework

In several industries where the CSPUL is present, the practical problem can be seen as closer to a multi-stage process. In this work, we propose a two-stage mathematical programming model that incorporates the effects of unknown future planning horizons (here modelled as the second stage) in the current period (the first stage).

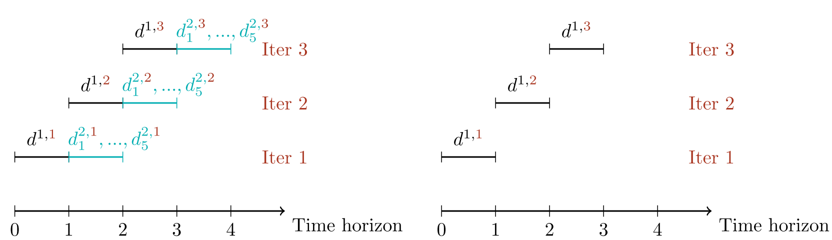
This section presents computational experiments that mimic a practical setting of consecutive periods or stages. These experiments demonstrate the value of this approach on a multi-period framework, solved on a rolling horizon. Figure 3 represents this experimental framework. For the stochastic problem (Fig. 3a), at each iteration *it*, the two-stage stochastic problem is solved considering the current period as the first stage, i.e., with known demand ($d^{1,it}$ in the figure) and the following as the second stage (i.e., with unknown demand, represented by scenarios, with potential outcomes $d_1^{2,it}, \ldots, d_5^{2,it}$). The first-stage results of a given iteration, such as

Table 10
EVPI and VSS results for each instance group, aggregated according to the initial leftover stock.

	EVPI			VSS		
Instance quartile	BC	DC	DE	BC	DC	DE
1	985.9	923.9	972.2	4,899.4	2,550.8	6,143.2
2	1,999.3	1,833.2	1,828.2	6,102.3	2,433.6	7,751.0
3	3,887.2	3,503.9	3,723.6	12,176.8	5,658.2	15,480.4
4	2,449.4	1,968.6	2,568.8	8,700.3	3,986.5	11,375.1
Average	**2,330.4**	**2,057.4**	**2,273.2**	**7,969.7**	**3,657.3**	**10,187.4**

Table 11
Realised demand for the ten periods of the multi-period experimental framework (first-stage demand at each iteration).

	Time periods									
Item	1	2	3	4	5	6	7	8	9	10
1	108	23	119	399	137	354	316	395	87	382
2	257	71	128	163	319	169	136	177	180	156
3	321	101	123	117	392	121	122	133	270	131
4	319	137	125	132	373	105	102	111	262	111
5	270	176	126	105	337	117	95	102	224	109
6	225	211	119	97	287	114	89	112	193	105
7	175	188	108	93	208	90	85	104	136	111
8	125	192	120	101	157	93	96	101	97	106
9	90	214	133	104	104	88	75	114	64	90
10	63	191	111	118	56	102	73	98	37	93
11	25	165	119	110	30	103	96	109	17	100
12	9	139	108	121	10	104	91	126	11	105
13	2	138	114	132	6	132	124	137	1	123
14	2	71	109	178	1	164	146	183	0	167
15	0	24	118	365	0	349	336	394	0	355
Total demand	1991	2041	1780	2335	2417	2205	1982	2396	1579	2244



(a) Two-stage stochastic approach (b) Deterministic approach

Fig. 3. Multi-period experimental framework, with $d_c^{it,st}$ for each iteration *it* as the demand in stage *st* (and scenario *c*, for the second stage).

the leftovers generated on the first stage, will be the inputs and starting point of the following iteration. The period that was previously considered the second-stage is now the first-stage. The process continues for a number of iterations, which was set to ten in these experiments.

For this multi-period setting, the demand values for each item were generated following a similar approach to the one previously described, based on Monte Carlo simulation and the beta distribution. For each iteration, two types of demand values were generated: the first-stage demand and the second-stage demand scenarios.

The total first-stage demand (summed up for all items) was randomly generated following an uniform distribution between 1500 and 2500. Then, one of the five demand profiles described in Table 4 was randomly selected and, based on the procedure described in Section 4, the total demand was distributed by the different items according to that profile. The resulting values that represent the actual realised demand for each period of the rolling-horizon framework are presented in Table 11.

Five scenarios are generated to represent the second-stage demand, each following one of the demand profiles presented in Table 4, as in previous experiments. For each scenario, the total demand (summed up for all items) was randomly generated following an uniform distribution between 1500 and 2500 (similarly to the first-stage demand). Based on the procedure described in Section 4, the total demand was then distributed by the different items according to the demand profile of each scenario. For simplicity, all scenarios are considered to be equally likely to occur (base case described in Tables 14, 15 and 16 in the Appendix present the values of demand for all scenarios and iterations. All data for the multi-period experimental setting is also available in the Supplementary Materials (Tables C1 and C2).

From this experimental setting results a situation that is similar to realistic industrial problems, as we are not assuming that when uncertainty is realised it is exactly as one of the estimated scenarios. Instead, the following iteration may reveal that the (certain) demand does not correspond exactly to one of the previous iteration's scenarios, but follows a foreseeable and approximated distribution and dimension.

As a benchmark of our stochastic approach, a deterministic cutting-stock problem with usable leftovers was used. In each iteration, which in this case corresponds to considering a single pe-

Table 12
Total first-stage cost (sum over all iterations) and cost decrease of the stochastic vs deterministic approach.

	Total first-stage cost (obj. value)		
Instance	Deterministic approach	Stochastic approach	Decrease
1	9,770,000.0	9,665,458.4	-1.07%
2	9,886,000.0	9,730,966.5	-1.57%
3	7,526,000.0	7,481,599.7	-0.59%
4	10,398,000.0	10,379,333.2	-0.18%
5	8,089,000.0	8,061,644.7	-0.34%
6	8,538,000.0	8,292,210.7	-2.88%
7	8,573,000.0	8,461,728.3	-1.30%
8	11,871,000.0	11,538,241.4	-2.80%
9	9,459,000.0	9,392,786.1	-0.70%
10	9,878,000.0	9,134,749.1	-7.52%
		Average	**-1.90%**

riod, the (known) demand of the period is fulfilled, either by using standard objects or leftovers in stock, and new leftovers may be generated and inserted in the stock for the subsequent periods (Fig. 3b). Technically, the same column generation procedure is used, but only with the first-stage variables and constraints and without generating columns regarding the second stage.

In this multi-period setting, it is mandatory to have integer solutions so that the relevant information fed to the following periods is feasible. Therefore, a restricted master heuristic (i.e. solving the MIP version of the problem, restricted to the variables generated by the column generation process) was used, both for the stochastic and deterministic approaches.

The multi-period demand setting was run for 10 out of the 50 instances described in Section 4, which are available in the Supplementary Materials (Tables C3, C4 and C5) representing different types of items, standard objects and leftovers.

Table 12 presents the total cost (derived from the objective function) across all iterations of the first-stage decisions. It should be noted that the first-stage cost represents the actual cost at a given period for the company, whereas the second-stage cost represents, at each iteration, an expectation of the future (which will be realised in the first-stage of the following iteration). Therefore, this table estimates the real value of applying a stochastic approach. The full detailed results of these experiments are available in Table 17 in the Appendix.

6. Conclusions

In this paper, we proposed a study of the CSPUL considering that the demand for items can be approximated by employing a finite set of scenarios in a certain sample space. We approached this problem as a two-stage stochastic program with recourse where each stage consists of a specific group of decision variables. We proposed a mixed-integer programming model to represent the problem and developed a column generation approach to solve it.

We generated fifty random instances for this problem, as well as a representative set of scenarios representing different demand profiles. This information is made available to support future research on this problem. Computational experiments validated the efficiency of the proposed approach and allowed us to understand the impact of different demand profiles, the role of uncertainty in this problem, the relevance of methods that explicitly tackle uncertainty, and the effect of varying probability distributions.

Overall, in this work, we aimed to show that the use of leftovers intrinsically requires the structure of the problem to consider more than one decision-making moment. As it is possible to generate leftovers to be later used to cut items, this multi-period structure is latent and can be explicitly used to derive better results. The literature in this field is starting to move towards this structure with, e.g. multi-stage deterministic approaches. In this work, we took a step forward by proposing a model and exact solution method that considered uncertainty. This makes this problem even more realistic, as the second stage of decision-making represents the uncertain future. We showed that considering uncertainty when modeling and solving the CSPUL significantly impacts the overall savings and the environmentally heavy waste generation.

From a technological point of view, this research is timely as technologies underlying the “Industry 4.0” framework, in which the Internet of Things (IoT) plays a critical role, are mature and starting to be widely used. Therefore, more and more data will be available, both in quantity and quality, allowing for the development of more accurate scenarios and the rigorous tracing of leftovers in stock. Indeed, the added management cost will be almost null, and the benefits are significant, as it is shown in this paper.

Acknowledgments

This research was sponsored and funded by FAPESP - Fundação de Amparo á Pesquisa do Estado de São Paulo (2016/01860-1, 2018/07240-0 and 2015/03066-8), Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (421130/2018-0 and 317460/2021-8) and by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, I.P., within project POCI-01-0145-FEDER-029609. The authors would like to thank Professor Douglas Alem for fruitful discussions in the early stages of the work.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi: 10.1016/j.ejor.2022.11.013.

Appendix

Table 13
Complete results of the computational tests: final objective value (Eq. (12)) and computational time in seconds for each probability distribution case considered: base case (BC), demand on center (DC) and demand on extremes (DE).

	Objective value			Time (sec)		
Instance	BC	DC	DE	BC	DC	DE
1	2,057,278.5	2,019,924.3	2,077,340.0	973.2	625.7	700.0
2	2,020,873.9	2,001,409.5	2,032,657.9	528.7	364.4	401.2
3	1,541,314.9	1,532,112.7	1,548,086.8	1,155.3	1,507.8	1,644.6
4	2,115,694.3	2,108,553.1	2,120,043.9	393.1	480.3	417.4
5	1,649,478.7	1,646,069.5	1,653,224.2	1,119.3	1,379.6	1,380.3
6	1,793,252.1	1,770,453.9	1,805,024.5	1,378.6	1,430.7	1,615.4
7	1,751,148.4	1,736,867.2	1,761,720.8	1,363.6	1,186.2	1,259.7
8	2,423,381.4	2,434,022.6	2,417,714.1	454.7	349.1	397.7
9	1,898,053.3	1,877,448.7	1,911,176.5	687.7	593.0	779.7
10	1,997,804.9	1,960,524.5	2,022,736.8	870.7	915.4	870.4
11	1,800,878.7	1,787,141.8	1,809,267.1	1,412.0	927.8	1,255.2
12	1,556,635.3	1,541,020.6	1,568,725.9	1,658.8	1,615.7	661.1
13	1,987,417.4	1,972,228.0	1,997,297.2	883.7	890.2	785.0
14	2,043,383.1	2,009,535.6	2,062,027.8	855.4	952.4	721.7
15	1,861,253.4	1,833,620.7	1,877,827.4	1,785.9	739.1	1,265.1
16	2,340,000.3	2,335,468.4	2,340,520.5	432.1	288.0	462.0
17	1,742,388.4	1,719,932.7	1,753,615.6	1,339.9	852.1	1,570.4
18	1,913,540.7	1,879,963.5	1,933,513.4	999.1	631.7	1,094.1
19	1,784,518.8	1,762,272.1	1,796,477.4	1,086.2	698.8	1,247.2
20	1,557,273.2	1,550,570.7	1,563,977.8	1,813.0	1,024.0	1,954.0
21	1,720,346.8	1,712,568.5	1,724,999.6	1,225.1	858.1	1,616.7
22	1,812,313.6	1,777,552.1	1,830,313.3	1,138.6	754.4	1,336.5
23	1,732,987.5	1,707,915.6	1,747,398.2	1,447.7	894.0	1,540.9
24	1,393,802.5	1,385,056.0	1,401,534.7	1,507.0	1,008.5	1,690.6
25	1,493,317.8	1,475,885.1	1,506,888.4	1,820.1	1,343.1	2,026.3
26	1,727,487.8	1,699,721.4	1,743,510.1	1,159.3	1,231.3	1,144.3
27	1,887,168.3	1,855,446.0	1,906,288.3	857.7	642.6	684.4
28	1,629,341.7	1,631,293.8	1,628,460.3	944.2	1,408.1	668.1
29	1,729,858.1	1,722,189.8	1,737,580.4	913.9	1,349.7	742.6
30	1,918,196.3	1,871,914.1	1,947,019.3	957.7	1,099.6	967.7
31	2,606,472.6	2,623,719.9	2,591,366.0	282.5	324.5	350.8
32	1,860,787.0	1,820,774.8	1,884,394.5	985.4	1,265.1	1,382.6
33	1,663,617.9	1,658,430.1	1,667,550.9	1,052.6	1,323.4	1,223.4
34	1,680,920.8	1,675,317.1	1,688,891.2	1,543.6	1,376.7	731.6
35	1,693,953.2	1,686,194.0	1,700,101.3	1,439.9	1,071.6	2,418.6
36	1,559,586.3	1,552,273.7	1,566,755.2	1,445.4	1,377.3	1,674.1
37	1,695,773.8	1,697,986.1	1,693,941.5	1,387.3	1,469.2	1,646.4
38	1,662,226.4	1,656,828.7	1,668,082.4	1,039.3	1,172.5	1,249.4
39	1,395,796.6	1,391,609.4	1,399,581.5	2,414.0	1,470.3	3,536.8
40	1,806,393.9	1,788,539.7	1,816,437.4	1,349.2	1,320.4	2,062.0
41	1,831,237.5	1,803,868.5	1,846,944.9	1,308.0	1,290.6	1,725.0
42	1,729,219.5	1,732,464.8	1,725,589.6	1,623.0	1,689.6	1,974.8
43	1,384,306.1	1,367,515.0	1,398,365.4	1,331.5	1,332.0	1,776.8
44	1,571,829.3	1,571,819.6	1,573,966.5	1,799.8	1,695.1	2,869.4
45	1,721,058.3	1,715,584.2	1,724,446.4	1,591.6	1,718.4	1,691.4
46	1,391,395.4	1,385,772.9	1,397,372.0	1,275.6	1,380.6	1,341.9
47	1,725,004.0	1,702,415.9	1,738,588.3	1,197.9	1,322.5	1,453.2
48	1,655,400.4	1,649,872.3	1,659,564.9	1,130.9	1,233.8	1,203.7
49	1,907,530.1	1,912,863.4	1,902,064.7	461.0	446.3	446.4
50	1,894,103.4	1,879,038.0	1,903,449.1	844.6	893.0	938.3
Average	**1,786,340.1**	**1,771,831.4**	**1,795,488.4**	**1,173.3**	**1,064.3**	**1,291.9**

Table 14
Second-stage total demand in each iteration and scenario of the multi-period experimental framework.

	Scenario				
Iteration	1	2	3	4	5
1	1519	1918	2332	1529	1815
2	1837	1998	1925	1726	2190
3	2339	1589	1679	2161	2231
4	1504	1939	2286	2226	2215
5	1764	1998	2052	1822	2229
6	2069	2032	1634	2230	2020
7	1633	2165	1826	2272	1778
8	1876	1734	2228	2357	1938
9	1561	1723	1969	1894	2391
10	2401	1504	1736	1653	1942

Table 15
Second-stage demand for each item in each iteration (it.) and scenario (scen.) of the multi-period experimental framework.

		Items														
It.	Scen.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	**1**	82	196	245	244	206	172	134	95	69	48	19	7	1	1	0
1	**2**	0	1	0	10	21	46	67	116	167	248	267	307	300	260	108
1	**3**	155	167	162	164	166	156	142	157	174	145	155	142	149	143	155
1	**4**	14	55	80	126	122	145	143	150	160	146	133	100	74	56	25
1	**5**	311	128	102	97	68	85	81	81	73	82	75	83	110	121	318
2	**1**	98	227	297	320	260	217	166	110	75	40	18	6	2	1	0
2	**2**	0	2	2	12	23	42	76	119	167	255	293	306	325	261	115
2	**3**	125	127	132	132	121	118	133	128	134	125	135	133	130	133	119
2	**4**	25	60	76	115	155	175	176	171	162	167	140	125	90	65	24
2	**5**	372	152	128	109	106	102	109	104	88	90	98	103	120	163	346
3	**1**	129	290	378	381	344	264	208	150	94	54	35	10	2	0	0
3	**2**	0	1	1	8	18	39	65	106	149	174	226	244	240	224	94
3	**3**	126	124	112	114	139	109	97	111	97	107	94	115	108	116	110
3	**4**	30	72	118	170	173	198	223	228	203	201	174	154	117	76	24
3	**5**	372	149	139	108	110	94	97	94	79	106	99	114	131	169	370
4	**1**	86	193	244	235	199	199	133	99	65	29	16	5	1	0	0
4	**2**	0	0	3	8	27	47	70	123	187	235	257	313	317	247	105
4	**3**	163	152	167	133	143	152	146	150	159	153	165	139	160	159	145
4	**4**	24	88	120	167	214	192	220	201	230	202	192	152	119	78	27
4	**5**	367	155	122	123	98	87	91	100	92	117	113	117	122	145	366
5	**1**	99	231	267	290	253	202	158	113	69	45	21	10	5	1	0
5	**2**	0	0	3	12	26	51	80	128	180	246	280	327	297	244	124
5	**3**	137	152	144	144	140	136	139	146	144	134	120	122	131	135	128
5	**4**	24	62	102	135	154	166	189	166	162	156	155	139	114	74	24
5	**5**	358	175	122	105	97	95	96	92	91	94	105	111	112	158	418
6	**1**	126	283	316	336	299	238	188	125	77	49	25	5	2	0	0
6	**2**	0	0	3	13	27	46	72	113	201	242	285	327	345	271	87
6	**3**	107	109	99	120	98	117	108	112	105	100	119	109	121	101	109
6	**4**	31	75	109	157	193	220	220	224	211	215	194	159	122	73	27
6	**5**	323	153	137	101	115	78	85	86	84	83	90	103	106	133	343
7	**1**	95	213	256	263	238	202	142	102	52	38	22	8	2	0	0
7	**2**	0	0	4	9	20	59	87	139	186	268	296	380	338	258	121
7	**3**	115	116	112	132	129	123	130	144	127	127	125	97	113	125	111
7	**4**	36	89	118	155	191	211	225	249	215	215	167	164	132	79	26
7	**5**	295	133	99	80	78	91	78	75	78	75	82	95	96	124	299
8	**1**	101	231	301	299	260	230	179	124	79	37	23	9	3	0	0
8	**2**	0	1	2	6	17	39	70	97	162	203	255	313	266	202	101
8	**3**	164	142	151	137	160	153	152	152	145	149	134	149	152	148	140
8	**4**	31	92	141	163	202	232	225	230	225	214	200	161	133	81	27
8	**5**	299	161	103	108	85	90	92	81	76	78	105	110	106	113	331
9	**1**	73	198	244	251	233	195	140	98	67	34	21	5	1	1	0
9	**2**	0	1	2	8	13	38	86	112	153	196	251	277	265	219	102
9	**3**	140	143	137	117	135	120	136	119	133	125	160	127	121	130	126
9	**4**	23	78	98	124	155	174	197	199	196	192	141	125	115	52	25
9	**5**	389	175	146	107	112	118	120	94	98	96	117	119	135	176	389
10	**1**	130	314	367	418	340	279	211	147	94	65	22	7	6	1	0
10	**2**	0	0	2	6	22	45	55	90	136	173	214	252	235	190	84
10	**3**	139	122	107	102	117	119	125	117	106	105	114	119	99	126	119
10	**4**	24	61	97	116	147	155	165	168	165	134	129	115	88	67	22
10	**5**	317	145	110	82	104	89	74	85	94	83	107	106	111	135	300

Table 16
Second-stage total demand in each iteration and scenario of the multi-period experimental framework.

	Scenario				
Iteration	1	2	3	4	5
1	1519	1918	2332	1529	1815
2	1837	1998	1925	1726	2190
3	2339	1589	1679	2161	2231
4	1504	1939	2286	2226	2215
5	1764	1998	2052	1822	2229
6	2069	2032	1634	2230	2020
7	1633	2165	1826	2272	1778
8	1876	1734	2228	2357	1938
9	1561	1723	1969	1894	2391
10	2401	1504	1736	1653	1942

Table 17
Full results for the multi-period computational experiments, with the stochastic and deterministic approaches, for all instances (Inst) and iterations (Iter).

		Objective value - Stochastic			Obj Value -
Inst	Iter	1st stage	2nd stage	Total	Deterministic
1	1	757,709.3	955,643.5	1,713,352.8	743,000.0
1	2	1,022,601.5	1,059,077.0	2,081,678.5	1,041,000.0
1	3	865,057.4	1,036,472.3	1,901,529.6	880,000.0
1	4	1,162,512.8	1,058,688.3	2,221,201.0	1,179,000.0
1	5	897,677.5	1,065,071.1	1,962,748.6	899,000.0
1	6	1,078,219.1	1,059,693.0	2,137,912.1	1,104,000.0
1	7	998,449.4	1,020,151.4	2,018,600.8	1,007,000.0
1	8	1,203,027.6	1,009,843.8	2,212,871.3	1,214,000.0
1	9	592,470.4	1,015,177.7	1,607,648.1	590,000.0
1	10	1,087,733.5	961,395.0	2,049,128.5	1,113,000.0
2	1	795,921.4	922,142.7	1,718,064.1	786,000.0
2	2	1,017,951.8	1,029,483.8	2,047,435.6	1,042,000.0
2	3	846,759.5	1,018,365.6	1,865,125.1	876,000.0
2	4	1,147,251.5	1,046,532.5	2,193,784.0	1,173,000.0
2	5	952,381.7	1,037,584.5	1,989,966.2	949,000.0
2	6	1,060,193.1	1,047,867.0	2,108,060.2	1,104,000.0
2	7	1,005,426.2	1,003,120.2	2,008,546.3	1,008,000.0
2	8	1,203,911.9	987,169.4	2,191,081.2	1,211,000.0
2	9	635,207.7	980,478.3	1,615,686.0	626,000.0
2	10	1,065,961.7	957,068.7	2,023,030.4	1,111,000.0
3	1	537,157.5	712,391.1	1,249,548.6	535,000.0
3	2	792,041.3	786,983.6	1,579,024.9	803,000.0
3	3	678,856.8	773,913.1	1,452,769.9	682,000.0
3	4	924,871.4	789,319.4	1,714,190.8	927,000.0
3	5	648,308.6	791,883.5	1,440,192.1	646,000.0
3	6	862,743.1	784,722.4	1,647,465.4	875,000.0
3	7	790,885.9	748,447.3	1,539,333.1	794,000.0
3	8	955,231.0	746,150.0	1,701,381.0	959,000.0
3	9	423,893.4	761,205.1	1,185,098.5	423,000.0
3	10	867,610.7	719,840.2	1,587,450.9	882,000.0
4	1	811,610.4	976,301.5	1,787,911.9	817,000.0
4	2	1,084,930.4	1,077,233.4	2,162,163.8	1,091,000.0
4	3	936,671.7	1,068,591.4	2,005,263.2	939,000.0
4	4	1,223,943.6	1,067,823.5	2,291,767.2	1,226,000.0
4	5	984,932.0	1,087,412.3	2,072,344.4	985,000.0
4	6	1,164,749.2	1,069,214.1	2,233,963.3	1,166,000.0
4	7	1,069,000.0	1,012,073.2	2,081,073.2	1,069,000.0
4	8	1,269,989.5	1,017,361.8	2,287,351.3	1,271,000.0
4	9	657,000.0	1,045,669.4	1,702,669.4	657,000.0
4	10	1,176,506.2	986,656.2	2,163,162.5	1,177,000.0
5	1	598,008.3	761,201.9	1,359,210.2	599,000.0
5	2	846,222.3	834,890.9	1,681,113.2	851,000.0
5	3	727,836.8	828,727.2	1,556,564.0	736,000.0
5	4	982,498.5	835,842.4	1,818,341.0	983,000.0
5	5	721,741.4	848,267.7	1,570,009.1	721,000.0
5	6	919,532.8	832,258.7	1,751,791.5	926,000.0
5	7	844,820.5	792,325.8	1,637,146.2	845,000.0
5	8	1,014,425.5	796,085.1	1,810,510.6	1,016,000.0
5	9	472,765.5	817,768.7	1,290,534.2	473,000.0
5	10	933,793.2	767,853.9	1,701,647.2	939,000.0
6	1	592,154.4	834,762.2	1,426,916.6	588,000.0
6	2	909,521.8	914,714.6	1,824,236.4	917,000.0
6	3	770,620.2	887,309.1	1,657,929.3	776,000.0
6	4	1,018,730.2	926,902.5	1,945,632.7	1,076,000.0
6	5	710,758.1	922,182.3	1,632,940.3	706,000.0
6	6	942,940.5	920,245.3	1,863,185.9	998,000.0
6	7	868,068.2	901,078.7	1,769,146.9	910,000.0
6	8	1,050,814.2	910,646.0	1,961,460.2	1,104,000.0
6	9	468,341.2	883,320.2	1,351,661.4	462,000.0
6	10	960,261.9	827,699.3	1,787,961.2	1,001,000.0
7	1	596,183.2	830,237.2	1,426,420.4	591,000.0
7	2	891,478.1	910,206.5	1,801,684.6	918,000.0
7	3	778,118.4	887,385.5	1,665,503.9	791,000.0
7	4	1,049,468.9	925,808.4	1,975,277.3	1,069,000.0
7	5	715,828.3	920,281.4	1,636,109.7	712,000.0
7	6	983,868.0	903,142.2	1,887,010.2	997,000.0
7	7	895,301.5	875,739.0	1,771,040.5	916,000.0
7	8	1,086,752.3	863,424.6	1,950,176.9	1,107,000.0
7	9	465,771.0	873,569.7	1,339,340.7	464,000.0
7	10	998,958.6	812,431.1	1,811,389.8	1,008,000.0
8	1	803,731.9	1,078,626.5	1,882,358.4	789,000.0
8	2	1,361,312.8	1,317,941.1	2,679,253.9	1,536,000.0

(*continued on next page*)

Table 17 (*continued*)

Inst	Iter	Objective value - Stochastic 1st stage	2nd stage	Total	Obj Value - Deterministic
8	3	1,161,000.0	1,268,533.6	2,429,533.6	1,161,000.0
8	4	1,418,768.1	1,319,009.9	2,737,778.0	1,422,000.0
8	5	972,136.6	1,239,496.7	2,211,633.3	951,000.0
8	6	1,220,409.6	1,297,210.0	2,517,619.7	1,341,000.0
8	7	1,210,941.1	1,270,339.8	2,481,280.9	1,211,000.0
8	8	1,480,000.0	1,250,161.1	2,730,161.1	1,481,000.0
8	9	638,325.8	1,211,575.2	1,849,901.0	623,000.0
8	10	1,271,615.5	1,182,311.1	2,453,926.6	1,356,000.0
9	1	760,814.4	865,683.5	1,626,497.9	752,000.0
9	2	967,647.4	956,325.7	1,923,973.1	975,000.0
9	3	825,649.9	949,675.8	1,775,325.7	837,000.0
9	4	1,117,809.6	956,988.1	2,074,797.8	1,121,000.0
9	5	901,954.3	968,870.8	1,870,825.1	911,000.0
9	6	1,049,455.0	957,101.6	2,006,556.5	1,063,000.0
9	7	958,260.5	911,013.7	1,869,274.2	964,000.0
9	8	1,159,603.7	908,577.9	2,068,181.5	1,160,000.0
9	9	586,316.1	927,629.3	1,513,945.4	599,000.0
9	10	1,065,275.2	876,573.1	1,941,848.3	1,077,000.0
10	1	680,966.4	902,680.0	1,583,646.4	670,000.0
10	2	993,582.9	1,022,358.7	2,015,941.6	1,136,000.0
10	3	813,904.1	1,049,398.8	1,863,302.9	933,000.0
10	4	1,157,303.1	1,157,623.3	2,314,926.4	1,230,000.0
10	5	819,761.7	998,446.2	1,818,207.9	807,000.0
10	6	994,154.7	1,019,837.3	2,013,992.0	1,136,000.0
10	7	905,413.8	1,066,280.0	1,971,693.8	1,037,000.0
10	8	1,207,547.0	1,080,100.0	2,287,647.0	1,263,000.0
10	9	552,985.4	940,621.1	1,493,606.5	526,000.0
10	10	1,009,130.0	901,618.1	1,910,748.2	1,140,000.0

References

Abuabara, A., & Morabito, R. (2009). Cutting optimization of structural tubes to build agricultural light aircrafts. *Annals of Operations Research, 169*(1), 149–165.

Alem, D. J., Munari, P. A., Arenales, M. N., & Ferreira, P. A. V. (2010). On the cutting stock problem under stochastic demand. *Annals of Operations Research, 179*(1), 169–186.

Arenales, M. N., Cherri, A. C., Nascimento, D. N. d., & Vianna, A. (2015). A new mathematical model for the cutting stock/leftover problem. *Pesquisa Operacional, 35*(3), 509–522.

Beraldi, P., Bruni, M. E., & Conforti, D. (2009). The stochastic trim-loss problem. *European Journal of Operational Research, 197*(1), 42–49.

Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

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.

Cherri, A. C., Arenales, M. N., & Yanasse, H. H. (2013). The usable leftover one-dimensional cutting stock problema priority-in-use heuristic. *International Transactions in Operational Research, 20*(2), 189–199.

Cherri, A. C., Arenales, M. N., Yanasse, H. H., Poldi, K. C., & Vianna, A. C. G. (2014). The one-dimensional cutting stock problem with usable leftovers–a survey. *European Journal of Operational Research, 236*(2), 395–402.

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.

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.

do Nascimento, D. N., de Araujo, S. A., & Cherri, A. C. (2020). Integrated lot-sizing and one-dimensional cutting stock problem with usable leftovers. *Annals of Operations Research*, 1–19.

Gilmore, P. C., & Gomory, R. E. (1963). A linear programming approach to the cutting stock problemPart II. *Operations Research, 11*(6), 863–888.

Gradišar, M., Jesenko, J., & Resinovič, G. (1997). Optimization of roll cutting in clothing industry. *Computers & Operations Research, 24*(10), 945–953.

Khan, R., Pruncu, C. I., Khan, A. S., Naeem, K., Abas, M., Khalid, Q. S., & Aziz, A. (2020). A mathematical model for reduction of trim loss in cutting reels at a make-to-order paper mill. *Applied Sciences, 10*(15), 5274.

Melhem, N. N., Maher, R. A., & Sundermeier, M. (2021). Waste-based management of steel reinforcement cutting in construction projects. *Journal of Construction Engineering and Management, 147*(7), 04021056.

Ravelo, S. V., Meneses, C. N., & Santos, M. O. (2020). Meta-heuristics for the one-dimensional cutting stock problem with usable leftover. *Journal of Heuristics*, 1–34.

Scheithauer, G. (1991). A note on handling residual lengths. *Optimization, 22*(3), 461–466.

Sculli, D. (1981). A stochastic cutting stock procedure: Cutting rolls of insulating tape. *Management Science, 27*(8), 946–952.

Tomat, L., & Gradišar, M. (2017). One-dimensional stock cutting: Optimization of usable leftovers in consecutive orders. *Central European Journal of Operations Research*, 1–17.

Trkman, P., & Gradisar, M. (2007). One-dimensional cutting stock optimization in consecutive time periods. *European Journal of Operational Research, 179*(2), 291–301.