Applied Mathematical Modelling 48 (2017) 250–268

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Decomposition methods for the lot-sizing and cutting-stock problems in paper industries

Aline A.S. Leao[a,*], Marcos M. Furlan[b], Franklina M.B. Toledo[a]

[a] *Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, Caixa Postal 668, 13560-970 São Carlos, SP, Brazil*
[b] *Universidade Federal de Grande Dourados, Faculdade de Ciências Exatas e Tecnológicas, Caixa Postal 364, 79804-970 Dourados, MS, Brazil*

ARTICLE INFO

Article history:
Received 9 February 2016
Revised 25 March 2017
Accepted 3 April 2017
Available online 7 April 2017

Keywords:
Cutting problem
Lot-sizing problem
Dantzig–Wolfe decomposition
Column generation method
Adaptive large neighborhood search

ABSTRACT

We investigate the one-dimensional cutting-stock problem integrated with the lot-sizing problem in the context of paper industries. The production process in paper mill industries consists of producing raw materials characterized by rolls of paper and cutting them into smaller rolls according to customer requirements. Typically, both problems are dealt with in sequence, but if the decisions concerning the cutting patterns and the production of rolls are made together, it can result in better resource management. We investigate Dantzig–Wolfe decompositions and develop column generation techniques to obtain upper and lower bounds for the integrated problem. First, we analyze the classical column generation method for the cutting-stock problem embedded in the integrated problem. Second, we propose the machine decomposition that is compared with the classical period decomposition for the lot-sizing problem. The machine decomposition model and the period decomposition model provide the same lower bound, which is recognized as being better than the linear relaxation of the classical lot-sizing model. To obtain feasible solutions, a rounding heuristic is applied after the column generation method. In addition, we propose a method that combines an adaptive large neighborhood search and column generation method, which is performed on the machine decomposition model. We carried out computational experiments on instances from the literature and on instances adapted from real-world data. The rounding heuristic applied to the first column generation method and the adaptive large neighborhood search combined with the column generation method are efficient and competitive.

1. Introduction

The one-dimensional cutting-stock problem (1CSP) and the lot-sizing problem (LSP) have been extensively studied over the past few decades. In the 1CSP, a set of objects is given to be cut into required items, where only one dimension of items and objects is relevant. On the other hand, the LSP determines how much to produce of a set of objects in order to satisfy the required demand in each period of a planning horizon without exceeding capacity constraints. Both problems are encountered in the paper, steel, glass and furniture industries. Although they are related to the same practical applications, there are few studies in the literature that address the interaction between these problems.

* Corresponding author.
E-mail addresses: aasleao@gmail.com (A.A.S. Leao), marcosfurlan@ufgd.edu.br (M.M. Furlan), fran@icmc.usp.br (F.M.B. Toledo).

http://dx.doi.org/10.1016/j.apm.2017.04.010

In this paper, we discuss the 1CSP integrated with the LSP in paper industries (1CSP–LSP). The production process consists of producing raw materials, i.e., rolls of paper and cutting them into final items in order to meet customer demands. Moreover, the required items are of different grades and different lengths. The production of raw materials may vary depending on the activities of each industry, in which the pulp can be purchased or produced by the industry itself. The pulp and different types of recycled paper are processed in paper machines. The industry may have one or more machines. After producing different grades of rolls of paper, two types of decisions must be considered: (i) cutting the rolls into the required items; (ii) stocking the items and rolls for the next period in the planning horizon.

The LSP is related to the production of raw materials. It should be determined which grades will be produced in each paper machine with a limited capacity during each period of the planning horizon. In addition, there are setup times and setup costs when changing the production of rolls with different grades. Some industries use higher quality grades to satisfy the demand of lower quality grades, which can avoid setups and paper waste. This strategy saves setup costs, but implies in additional production costs related to higher paper quality sold at lower prices. However, in this study we do not adopt this strategy, i.e., the grade of the rolls must meet the grade of the customer demand. Thus, the LSP considered in the combined problem, addressed in this paper, is the capacitated lot-sizing problem in parallel machines with setup times and setup costs (CLSPP). In each period of the planning horizon, the 1CSP determines the number of required items to be cut from the rolls of paper. The objective of the CLSPP is to minimize the production costs, setup costs and the quantity of rolls held at the end of each period. In the 1CSP, the objective is to minimize the paper wastage incurred during the cutting phase and the storage of items. The 1CSP–CLSPP integration occurs due to the fact that it is necessary to produce a sufficient quantity of rolls to be cut in order to meet the demand of items in each period of the planning horizon.

A strategy used in the literature to deal with the 1CSP–CLSPP consists of integrating the classical cutting stock model proposed by Gilmore and Gomory [1] and the classical lot-sizing model, where sequence dependent setup costs and times are not considered (see [2,3]). A simple way of handling this model (pattern-oriented model) is to solve its linear relaxation by a column generation method, where the subproblems generate only the cutting patterns. Then, the integer linear programming model with the columns available at the end of the column generation method is solved by a mixed integer programming (MIP) solver. The disadvantage of this strategy is that the resultant integer linear programming model is hard to solve when considering real-world data. To deal with this issue, we propose two decompositions to reduce the number of variables in the pattern-oriented model.

In order to describe a sequence of decompositions, four integer linear programming models are analyzed. The first model (item-oriented model – IOM) integrates the classical CLSPP formulation and the mathematical formulation of Kantorovich [4] for 1CSP. As previously described, in the second model (pattern-oriented model – POM), the 1CPS is modeled according to Gilmore and Gomory [1], while retaining the initial CLSPP formulation of the first model. In the third model (machine decomposition model – MDM), we consider the 1CSP model written according to the approach proposed by Gilmore and Gomory and we propose the Dantzig–Wolfe decomposition per machine for the CLSPP. The Dantzig–Wolfe decomposition per period is also considered (period decomposition model – PDM), but only briefly described.

Although the setup times and setup costs are sequence dependent, dealing with them is not within the scope of this paper. We consider the average setup values for each grade type. However, a post-processing procedure can be used to deal with the setup sequence. We highlight that in some industrial practices, the production manager works in this way. In addition, by considering the sequence dependent setup we need additional binary variables that change the structure of the analyzed models and, consequently, the PDM cannot be applied to solve the problem.

To solve the proposed models, we implemented column generation methods. First, lower bounds are determined by solving the linear relaxation of the models using the column generation methods. Then, we obtain feasible solutions (upper bounds) by imposing the integrality to the variables of the resultant restricted master problem and solving it by a MIP solver (rounding heuristic – RH). A further contribution of the paper consists of combining the column generation method performed on MDM with an adaptive large neighborhood search (CG–ALNS). At the end of the method, the resultant integer restricted master problem is also solved by a MIP solver. We evaluate the performance of the methods using the instances proposed by Poltroniere et al. [2] and instances adapted from real-world data from [5]. Computational results show that solving the integer restricted master problem of the POM by CPLEX 12.6 provides high-quality solutions in a low computational time for instances from [2]. The CG–ALNS heuristic can also provide almost-optimal solutions for these instances. Despite presenting a better solution for some of them, the average gap of CG–ALNS is 0.13% worse than the results obtained by the rounding heuristic performed in the POM. However, the CG–ALNS heuristic outperforms the rounding heuristics in all instances adapted from real-world data.

The main contributions of this paper are: (i) proposing of mathematical formulations for the 1CSP–CLSPP; (ii) comparing the lower and upper bounds obtained by solving the models; and (iii) proposing a heuristic method that combines the column generation method and the adaptive large neighborhood search (CG–ALNS). Different from previous developments concerning this problem, we combine the column generation method performed on a new mathematical formulation with a meta-heuristic method, where the proposed heuristic is compared with classical column generation heuristics.

The paper is organized as follows. In Section 2, we review some papers in the literature focused on integrating two or more problems in paper mill manufacturing. In Section 3, we present the mathematical models for the 1CSP–CLSPP. Section 4 describes the rounding heuristic and the adaptive large neighborhood search combined with the column generation method. The computational results are reported in Section 5. In Section 6, some conclusions are drawn.

2. Literature review

The production process in paper industries is complex and can involve many problems, such as production planning, scheduling problems, transportation problems, as well as one- and two-dimensional cutting-stock problems. There are only a few papers in the literature that solve the integration of two or more of these problems. Planning and scheduling problems involve decisions about how much paper will be produced in one or more machines during a planning horizon and the sequence of the paper types in each machine. Usually there are setup costs and setup times for changing from one type of paper to another one, where a paper type can be characterized by its grade and its color. After producing the rolls of paper, the rolls are cut into smaller ones so that material wastage is minimized, which characterizes the one-dimensional cutting-stock problem. The smaller rolls may be required items or intermediate products. When the required items are sheets, a two-dimensional cutting-stock problem is defined. Finally, the end products are delivered to customers.

Krichagina et al. [6] combined the production scheduling and the cutting-stock problem in a paper industry that produces sheets of different sizes. The production scheduling problem involves inventory decisions such as the production and inventory levels of items in order to determine planned shutdowns for the paper machine. The objective is to minimize backorders, holding costs, shutdowns and paper waste. Shutdowns in a paper machine are allowed since the machine capacity is slightly higher than the demand. To solve the problem, they proposed a linear programming model to decide a set of cutting configurations to be used in a Brownian analysis that identifies dynamic scheduling.

Keskinocak et al. [7] developed a support system for paper mill industries in different locations. Their support system considers distributing the demand among the paper mills, scheduling paper machines in each paper mill, cutting rolls into smaller ones (1CSP), and transporting the requested items to customers. To solve the integration of these problems, they used a series of algorithms in an asynchronous team framework.

Menon and Schrage [8] integrated the 1CSP with the assignment problem of rolls for identical cutting machines. They explored the dual-angular structure of the problem to decompose it and to obtain tight bounds for the subproblems that are used to solve a relaxation of the master problem.

Respicio et al. [9] also presented a support system for a paper mill company, which interacts with the decision makers. They integrated the capacity planning and scheduling of the paper machine with the 1CSP. The problem is decomposed into two subproblems. The first one determines the rolls to be produced over a discrete planning horizon and the second one defines the machine batching and scheduling. To define the cutting patterns, they used an integer-programming model based on [1]. The linear relaxation of the model is solved by the column generation method and a feasible solution is obtained by rounding the optimal linear relaxation solution. When the paper machine capacity is not completely used, the system computes the forecast demand based on records over 10 years. In addition, they accept loss of demand if the machine capacity is insufficient to produce all the required items.

Correia et al. [10] integrated the production planning and the cutting-stock problem, where they focus mainly on the generation of cutting patterns. The planning production involves decisions of how much paper must be produced to meet the demand. Due to technical characteristics, the rolls are cut into two stages: the first stage of cuts generates intermediate rolls and the second one generates the final rolls. As the required items may be rolls or sheets, some rolls need to be cut according to a two-dimensional cutting pattern. To solve the problem, some cutting patterns are enumerated and used as columns in a production-planning model.

Poltroniere et al. [2] joined the 1CSP and CLSPP applied to the paper industry. The problem consists of determining cutting patterns, deciding how to use them during a planning horizon and defining a production schedule for parallel paper machines. The objective is to minimize storage costs of rolls and final items, production and setup costs and paper waste generated during the cutting process. They presented an integer programming formulation and proposed two decomposition heuristics based on Lagrangian relaxation to solve the problem. In [3], a column generation method is applied to the linear relaxation of this model, where in each iteration of the simplex method the percentage of waste material needs to be estimated. At the end of the method, the integer restricted master problem is solved by a MIP solver, but the number of instances solved is smaller than the best heuristic of Poltroniere et al. [2]. The authors also described a model that combines the arc flow model for the cutting stock problem proposed by Valério de Carvalho [11] with the lot-sizing problem which is solved by a MIP solver. However, this model does not provide good results.

Correia et al. [12] extended the problem described in [10] by considering multiple machines for the paper production. To solve the problem, they used the model proposed in [8] to combine the assignment and cutting problems. The grade sequencing in each paper machine was defined according to the traveling salesman problem. However, the capacity constraint of the machines was ignored. In an attempt to ensure a feasible solution, they used a re-assignment heuristic.

Kim et al. [13] presented a mathematical formulation for the one-dimensional two-staged cutting-stock problem, where rolls of paper are cut into auxiliary rolls. To determine the auxiliary rolls, they defined the subpattern concept, where a subpattern consists of one or more auxiliary rolls of the same order. After the one-dimensional cut, the subpatterns can be cut into sheets. In the mathematical formulation, all subpatterns and patterns are known a priori. In order to solve the problem, they proposed a heuristic to generate the subpatterns and patterns, and a heuristic to select the patterns.

The integration of the cutting-stock problem and production planning in other industrial practices has also been studied in the literature such as in furniture [14–17], clothing [18], copper [19] and wood processing industries [20], steel truck manufacturing [21,22] and gear belts production [23].

3. Mathematical formulations for the 1CSP–CLSPP

In the 1CSP–CLSPP, we have to meet the demand in a planning horizon with T periods. The items are rolls of papers with different lengths and classified into K different grades. M machines are available to produce bigger rolls of papers (objects). The machines can have different sizes and capacities, then the length of the objects produced can vary according to the machine and the grade which is denoted by b_{km}, $k = 1, \ldots, K$, $m = 1, \ldots, M$. Hence, the objects must be cut into the required items of length given by l_i ($l_i \leq b_{km}$). There is a time factor for producing the objects and a setup time for changing the production of objects with different grades, where each machine has a capacity time. The problem consists of determining the production planning of the objects without exceeding the machine capacity and how they are cut. In addition, items and objects can be held from one period to another. The objective is to minimize production and setup costs, inventory costs of items and objects, as well as waste cost incurred during the cutting process.

In this section, we present three mathematical formulations for the 1CSP–CLSPP: the item-oriented model; the pattern-oriented model; and the machine decomposition model. For the mathematical formulations, we introduce the following sets, parameters and variables that are common for all models.

Sets:

- $\mathcal{T} = \{1, \ldots, T\}$: set of periods in the planning horizon
- $\mathcal{M} = \{1, \ldots., M\}$: set of machines
- $\mathcal{K} = \{1, \ldots, K\}$: set of object grade types
- $\mathcal{N} = \{1, \ldots n\}$: set of items, $\mathcal{N} = \mathcal{N}_1 \cup \cdots \cup \mathcal{N}_K$, where $\mathcal{N}_k$ is the set of items with grade of type k

Parameters:

- c_{km}: production cost of object type k in machine m, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$;
- stc_{km}: setup cost of machine m to produce object of type k, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$
- cl_{km}: paper waste cost of object type k produced in machine m, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$;
- $\bar{h}_{km}$: holding cost of object of type k produced in machine m at the end of each period, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$;
- h_i: holding cost of item of type i at the end of each period, $\forall i \in \mathcal{N}$;
- τ_i: additional holding cost of item of type i at the end of period T, $\forall i \in \mathcal{N}$;
- l_i: length of item of type i, $\forall i \in \mathcal{N}$;
- d_i^t: demand of items of type i in period t, $\forall i \in \mathcal{N}$, $\forall t \in \mathcal{T}$;
- b_{km}: length of object of type k produced in machine m, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$;
- Cap_m: capacity of machine m, $\forall m \in \mathcal{M}$;
- st_{km}: setup time of machine m to produce object of type k, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$;
- vt_{km}: production time of object k in machine m, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$;
- M_{km}: a big number defined as $M_{km} = \frac{Cap_m}{vt_{km}}$, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$.

Decision variables:

- e_{km}^t: number of objects of type k produced in machine m held at the end of period t, $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$, $\forall t \in \mathcal{T}$;
- s_i^t: holding quantity of item of type i at the end of period t, $\forall i \in \mathcal{N}$, $\forall t \in \mathcal{T}$;
- r_{km}^t: number of objects of type k produced in machine m in period t, $k \in \mathcal{K}$, $m \in \mathcal{M}$, $t \in \mathcal{T}$;
- z_{km}^t: 1 if object of type k is produced in machine m in period t, 0 otherwise, $k \in \mathcal{K}$, $m \in \mathcal{M}$, $t \in \mathcal{T}$.

3.1. Item-oriented model

In the item-oriented model (IOM), the 1CSP is modeled according to Kantorovich [4] which is combined with the classical CLSPP formulation. It is well-known that using commercial solvers to solve Kantorovich's model is impractical for large problems and it may even be challenging to store the mathematical model on a computer. Thus, the IOM is described only to illustrate a sequence of Dantzig–Wolfe decompositions that can be applied for 1CLSP–CLSPP. The Dantzig–Wolfe decomposition of the Kantorovich's model is equivalent to the model proposed by Gilmore and Gomory [1], which is introduced in the next section.

In the 1CSP, the number of objects necessary to meet the demand is unknown and, consequently, we must define an upper bound on the number of objects that can be used. The upper bound on the number of objects with grade k produced in machine m in period t is represented by O_{km}^t, where $\mathcal{O}_{km}^t = \{1, \ldots, O_{km}^t\}$ is the set of available objects. In this case, we must decide if an object o ($o \in \mathcal{O}_{km}^t$) is cut or not. A good estimation of O_{km}^t reduces the computational effort to solve and to store the mathematical model. Moreover, if an object is used, we must decide which items will be cut from it.

Let us consider the following additional variables.

- u_{okm}^t: 1 if object o of type k is produced in machine m and is cut in period t, 0 otherwise, $o \in \mathcal{O}_{km}^t$, $k \in \mathcal{K}$, $m \in \mathcal{M}$, $t \in \mathcal{T}$;
- x_{iom}^t: number of items of type i cut from object o of type k produced in machine m in period t, $i \in \mathcal{N}_k$, $o \in \mathcal{O}_{km}^t$, $k \in \mathcal{K}$, $m \in \mathcal{M}$, $t \in \mathcal{T}$.

Note that this modeling approach allows many symmetries since a cutting pattern might be associated with more than one object. Then, the item-oriented model is given as follows.

$$z^{IOM} = \text{minimize} \quad \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}} h_i s_i^t + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} \bar{h}_{km} e_{km}^t + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} stc_{km} z_{km}^t$$

$$+ \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_{km} r_{km}^t + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}}\sum_{o\in\mathcal{O}_{km}^t} cl_{km}\left(b_{km}u_{okm}^t - \sum_{i\in\mathcal{N}_k} l_i x_{iom}^t\right) + \sum_{i\in\mathcal{N}} \tau_i s_i^T, \tag{1}$$

$$\text{s.t.} \sum_{m\in\mathcal{M}}\sum_{o\in\mathcal{O}_{km}^t} x_{iom}^t + s_i^{t-1} = d_i^t + s_i^t \ \forall i\in\mathcal{N}, \quad \forall k\in\mathcal{K}, \forall t\in\mathcal{T}, \tag{2}$$

$$\sum_{i\in\mathcal{N}_k} l_i x_{iom}^t \le b_{km} u_{okm}^t \quad o\in\mathcal{O}_{km}^t \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}, \tag{3}$$

$$r_{km}^t + e_{km}^{t-1} - e_{km}^t = \sum_{o\in\mathcal{O}_{km}^t} u_{okm}^t \quad \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}, \tag{4}$$

$$\sum_{k\in\mathcal{K}} (st_{km} z_{km}^t + vt_{km} r_{km}^t) \le Cap_m \quad \forall m\in\mathcal{M}, \forall t\in\mathcal{T}, \tag{5}$$

$$r_{km}^t \le M_{km} z_{km}^t \ \forall k\in\mathcal{K}, \quad \forall m\in\mathcal{M}, t\in\mathcal{T}, \tag{6}$$

$$s_i^0 = 0, e_{km}^0 = 0 \ \forall i\in\mathcal{N}, \quad \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \tag{7}$$

$$x_{iom}^t, r_{km}^t, e_{km}^t, s_i^t \in \mathbb{Z}_+, z_{km}^t, u_{okm}^t \in \{0,1\} \ \forall i\in\mathcal{N}, o\in\mathcal{O}_{km}^t, \quad \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}. \tag{8}$$

The objective function (1) minimizes the sum of the holding cost of items (h_i), the holding cost of the objects ($\bar{h}_{km}$), the setup costs (stc_{km}), the production cost of objects (c_{km}), and the waste cost occurred in the cutting phase (cl_{km}). In addition, to avoid a high inventory of items in the last period, we penalized the high inventory of items by adding the term $\sum_{i\in\mathcal{N}} \tau_i s_i^T$ to the objective function. Constraints (2) are the operational demand constraints, where the inventory demand of items is considered. Constraints (3) determine feasible cutting patterns for the objects. The tactical demand constraints are represented in (4), they couple the cutting-stock problem with the lot-sizing problem by taking into account the total demand of objects for the cutting phase, and their production and holding stock. The production capacity constraint of the machines is given in (5). If an object is produced in period t in machine m, constraints (6) force its respective setup variable (z_{km}^t) to one. Without loss of generality, we assume in constraints (7) that there is no initial inventory. Constraints (8) define the domain of the variables.

The model proposed by Kantorovich [4] is well known for its weak linear relaxation, which is not different for the item-oriented model. Stronger relaxations can be obtained by the relaxation of the models described in the following sections.

3.2. Pattern-oriented model

Vance [24] showed that the Dantzig–Wolfe decomposition applied to the cutting stock model of Kantorovich [4] results in a model that is equivalent to the one proposed by Gilmore and Gomory [1]. In this section, we apply the Dantzig–Wolfe decomposition to the IOM in order to build an integrated model where the constraints related to the cutting problem are written according to the model proposed by Gilmore and Gomory [1]. To this end, let us define $\mathcal{Q}_{okm}^t$ as the set of extreme points of

$$conv\left\{\sum_{i\in\mathcal{N}_k} l_i x_{iom}^t \le b_{km}, x_{iom}^t \in \mathbb{Z}_+, i\in\mathcal{N}_k\right\},$$

for each object $o\in\mathcal{O}_{km}^t$, $\forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}$. We denote an extreme point by the vector $(a_{ij})_{i\in\mathcal{N}_k}$ for $j\in\mathcal{Q}_{okm}^t$. Each knapsack constraint (3) can be replaced by the convex combination of the extreme points in $\mathcal{Q}_{okm}^t$, where each extreme point is associated with a decision variable λ_{jokm}^t which is equal to 1 if the extreme cutting pattern j is performed in object o. Then, the new mathematical formulation is written as the following model referred to the master problem.

$$z = \text{minimize} \quad \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}} h_i s_i^t + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} \bar{h}_{km} e_{km}^t + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} stc_{km} z_{km}^t$$

$$+ \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_{km} r_{km}^t + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}}\sum_{o\in\mathcal{O}_{km}^t} cl_{km}(b_{km} - \sum_{i\in\mathcal{N}_k}\sum_{j\in\mathcal{Q}_{okm}^t} l_i a_{ij})\lambda_{jokm}^t + \sum_{i\in\mathcal{N}} \tau_i s_i^T. \tag{9}$$

s.t. (5)–(7)

$$\sum_{m\in\mathcal{M}}\sum_{o\in\mathcal{O}^t_{km}}\sum_{j\in\mathcal{Q}^t_{okm}} a_{ij}\lambda^t_{jokm} + s^{t-1}_i = d^t_i + s^t_i \quad \forall i\in\mathcal{N}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}, \tag{10}$$

$$\sum_{j\in\mathcal{Q}^t_{okm}} \lambda^t_{jokm} \le 1 \quad o\in\mathcal{O}^t_{km}, \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}, \tag{11}$$

$$r^t_{km} + e^{t-1}_{km} - e^t_{km} = \sum_{o\in\mathcal{O}^t_{km}}\sum_{j\in\mathcal{Q}^t_{okm}} \lambda^t_{jokm} \quad \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}, \tag{12}$$

$$r^t_{km}, e^t_{km}, s^t_i \in \mathbb{Z}_+, z^t_{km}, \lambda^t_{jokm} \in \{0,1\} \quad \forall i\in\mathcal{N}, o\in\mathcal{O}^t_{km}, \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}. \tag{13}$$

Constraints (10) require that the demand of each item is met by the chosen patterns. The convexity constraints (11) for each roll o ensure that only extreme pattern in the convex hull of the knapsack constraint can be chosen. Constraints (12) are the tactical demand constraint that are equivalent to constraints (4).

This model provides a stronger linear relaxation than IOM because the cutting patterns is restricted to be in the convex hull of integer solution of the knapsack problem. In addition, it can be simplified by changing the variables without eliminating any feasible integer solution. In this case, we consider not only the extreme points of the convex hull, but all feasible solutions (feasible cutting patterns) for the knapsack constraint of each object o:

$$\sum_{i\in\mathcal{N}_k} l_i x^t_{iom} \le b_{km},$$
$$x^t_{iom} \in \mathbb{Z}_+, \;\; i\in\mathcal{N}_k.$$

Thus, we can drop index o because all feasible solutions for the object of the same type k produced in machine m have the same set of feasible solutions. Moreover, the binary variable λ^t_{jkom} can be replaced by the integer variable y^t_{jkm} that represents the number objects of type k produced in machine m, in period t, and are cut according to pattern j. We denote the index set of cutting patterns for each k, m and t by $\mathcal{J}^t_{km} = \{1, \ldots, J^t_{km}\}$. Note that, since we are considering all feasible cutting patterns, $\mathcal{J}^1_{km} = \cdots = \mathcal{J}^T_{km}$ for each k and m. However, in order to describe the column generation method to solve the model after its definition, we also distinguish the set of cutting pattern according to the periods. Then, we obtain the following simplified version of model (9)–(13) referred to pattern-oriented model (POM):

$$\begin{aligned} z^{POM} = \text{minimize} \quad & \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}} h_i s^t_i + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} \bar{h}_{km} e^t_{km} + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} stc_{km} z^t_{km} \\ & + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} c^t_{km} r^t_{km} + \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}^t_{km}} cl_{km}\Big(b_{km} - \sum_{i\in\mathcal{N}_k} l_i a_{ij}\Big) y^t_{jkm} + \sum_{i\in\mathcal{N}} \tau_i s^T_i . \end{aligned} \tag{14}$$

s.t. (5)–(7)

$$\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{J}^t_{km}} a_{ij} y^t_{jkm} + s^{t-1}_i = d^t_i + s^t_i \quad \forall i\in\mathcal{N}_k, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}. \tag{15}$$

$$r^t_{km} + e^{t-1}_{km} - e^t_{km} = \sum_{j\in\mathcal{J}^t_{km}} y^t_{jkm} \quad \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}. \tag{16}$$

$$y^t_{jkm}, r^t_{km}, e^t_k, s^t_i \in \mathbb{Z}_+, z^t_{km} \in \{0,1\} \quad \forall i\in\mathcal{N}, j\in\mathcal{J}^t_{km}, \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall t\in\mathcal{T}. \tag{17}$$

This simplified model describes the cutting stock constraints according to the cutting stock model proposed by Gilmore and Gomory [1]. By comparing IOM and POM, it can be observed that each possible solution for variables x^t_{iokm} can be represented in a_{ij}, where a_{ij} is the number of items of type i cut according to pattern j, $i\in\mathcal{N}_k$, $j\in\mathcal{J}^t_{km}$, $m\in\mathcal{M}$, $k\in\mathcal{K}$, $t\in\mathcal{T}$. In POM, we overcome the symmetries of IOM as the decision variables determine how many objects are cut according to each pattern j. The integer variable y^t_{jkm} represents the number of objects of type k produced in machine m, in period t, and are cut according to pattern j, $j\in\mathcal{J}^t_{km}$, $k\in\mathcal{K}$, $m\in\mathcal{M}$, $t\in\mathcal{T}$. By defining the set of cutting patterns, the term $\sum_{o\in\mathcal{O}^t_{km}} u^t_{okm}$ in the IOM is equivalent to $\sum_{j\in\mathcal{J}^t_{km}} y^t_{jkm}$. In the IOM, a cutting pattern needs to be generated as often as it is required according to binary variable u^t_{okm}. Meanwhile, the POM decides how many objects will be cut according to a cutting pattern. In addition, the term $\sum_{o\in\mathcal{O}^t_{km}} x^t_{iom}$ is equivalent to $\sum_{j\in\mathcal{J}^t_{km}} a_{ij} y^t_{jkm}$.

This model is a variation of the one proposed by Poltroniere et al. [2]. The authors considered additional inventory balancing constraints for the objects, where the weight of the objects must meet the weight of the required items. In addition, the demand in these constraints is increased by a waste estimation. It is important to highlight that this estimation is necessary in their model since the problem was solved by heuristics that alternate between the 1CSP and the CLSPP solutions. Another difference between both models is the term $\sum_{i\in\mathcal{N}} \tau_i s^T_i$ added to the objective function to avoid a high inventory of items.

Since it might be computationally impractical to generate all the cutting patterns, they are considered implicit and the linear relaxation of both reformulated models can be solved by the column generation method. This method is used to solve linear programming models with a large number of columns, which is called the master problem. Instead of considering all columns at once, the model with only a subset of columns (restricted master problem) is solved by the simplex method, where in each iteration, new columns can be generated and added to the restricted master problem. The columns are provided by solving the pricing problems (subproblems). The method stops when there is no attractive column to be added to the restricted master problem, where the column is evaluated by its reduced cost.

In order to apply the column generation method to the POM, we consider only a subset of cutting patterns ($\bar{\mathcal{J}}_{km}^t \subseteq \mathcal{J}_{km}^t$) and the integrality of all variables in constraints (17) is relaxed. Additional attractive columns are added to the problem in each iteration of the simplex method by solving the subproblems (or pricing problems). A column is attractive in POM if its reduced cost is non-negative.

To determine the pricing problems, let us define π_i^t as the dual variables associated with constraints (15), $\forall i \in \mathcal{N}$, and $\forall t \in \mathcal{T}$, and σ_{km}^t the dual variables associated with constraints (16), $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$, $\forall t \in \mathcal{T}$. The reduced cost is given by:

$$rc1_{km}^t = cl_{km}b_{km} - \sum_{i \in \mathcal{N}_k} (\pi_i^t + cl_{km}l_i)a_i + \sigma_{km}^t,$$

that must be minimized. Then, the subproblem to be solved for the object of type k produced in machine m and in period t is formulated as follows.

$$\text{(Subproblem 1)} \quad \text{maximize} \sum_{i \in \mathcal{N}_k} (\pi_i^t + cl_{km}l_i)a_i, \tag{18}$$

$$\text{s.t.} \sum_{i \in \mathcal{N}_k} l_i a_i \le b_{km}, \tag{19}$$

$$a_i \in \mathbb{Z}_+, i \in \mathcal{N}_k. \tag{20}$$

In the column generation method, we have $K \times M \times T$ subproblems. Despite the previous observation of $\mathcal{J}_{km}^1 = \cdots = \mathcal{J}_{km}^T$ for each k and m, as the column generation method adds only the attractive columns to the master problem, the subsets of columns added to the master problem $\bar{\mathcal{J}}_{km}^1, \ldots, \bar{\mathcal{J}}_{km}^T$ are usually different. The difference between the subsets of cutting patterns for each k and m can be observed in the reduced costs $rc1_{km}^t$, which is due to de dual variables. If $rc1_{km}^t \ge 0$, for all k, m and t, then an optimal solution of the linear relaxation of the POM is found.

By solving the linear relaxation of the master problem, we only have a lower bound for the optimal solution of the POM. Feasible solutions for the master problem can be obtained by applying exact methods (e.g. branch-and-price method) or heuristic methods (e.g. rounding heuristics and constructive heuristics) to the restricted master problem. In Section 4, we describe the rounding heuristic applied to the restricted master problem obtained after optimally solving the linear relaxation of POM using the column generation method.

It can be clearly observed that the lower bound provided by the linear relaxation of the POM is not worse than the lower bound provided by the linear relaxation of the IOM. In the linear relaxation of the IOM, the decision variables of the number of items in the cutting patterns are relaxed, while in the POM they are kept as integer.

3.3. Machine decomposition model

In this section, we propose the Dantzig–Wolfe decomposition per machine for the lot-sizing constraints. The development of the period decomposition is very similar to the machine decomposition, therefore it is omitted here for brevity. The POM is decomposed into M subproblems and the master problem is rewritten, where each subproblem determines the production plan for one machine. For this, let us represent:

$$CM_m = conv\{\mathtt{z}_m \in \{0,1\}^{\mathcal{K}\times\mathcal{T}}, \mathtt{r}_m \in \mathbb{Z}_+^{\mathcal{K}\times\mathcal{T}} : \sum_{k \in \mathcal{K}} (st_{km}z_{km}^t + vt_{km}r_{km}^t) \le Cap_m, t \in \mathcal{T}; r_{km}^t \le M_{km}z_{km}^t, k \in \mathcal{K}, t \in \mathcal{T}\},$$

as the convex hull of the feasible solutions of constraints (5) and (6) for machine m. In this case, the master problem chooses the production plan for each machine and the cutting patterns according to the pattern-oriented model. Thus, we have $K \times M \times T$ subproblems related to the pattern generation and M subproblems related to the production plan. For the machine decomposition model (MDM), we define the following parameters and variables.

Parameters:

- $\mathcal{P}_m$: set of convex hull's vertices, i.e., a set of extreme production plans related to machine m, $m \in \mathcal{M}$;
- $r_{km}^{tp_m}$: number of objects of type k produced in machine m in period t according to production plan p_m, $k \in \mathcal{K}$, $t \in \mathcal{T}$, $p_m \in \mathcal{P}_m$, $m \in \mathcal{M}$;
- $z_{km}^{tp_m}$: 1 if object of type k is produced in machine m, in period t according to production plan p_m, 0 otherwise, $k \in \mathcal{K}$, $t \in \mathcal{T}$, $p_m \in \mathcal{P}_m$, $m \in \mathcal{M}$.

Variables:

- $\bar{r}_m^{p_m}$: 1 if the plan p_m of machine m is used, and 0 otherwise, $p_m \in \mathcal{P}_m$, $m \in \mathcal{M}$.

Then, the variables r_{km}^t and z_{km}^t can be rewritten as the convex combination of the vertices of CM_m, $\forall m \in \mathcal{M}$:

$$r_{km}^t = \sum_{p_m \in \mathcal{P}_m} r_{km}^{t\,p_m} \bar{r}_m^{p_m} \quad \forall k \in \mathcal{K}, \forall t \in \mathcal{T},$$

$$z_{km}^t = \sum_{p_m \in \mathcal{P}_m} z_{km}^{t\,p_m} \bar{r}_m^{p_m} \quad \forall k \in \mathcal{K}, \forall t \in \mathcal{T}$$

$$\sum_{p_m \in \mathcal{P}_m} \bar{r}_m^{p_m} \le 1$$

$$\bar{r}_m^{p_m} \in \mathbb{Z}_+ \quad \forall p_m \in \mathcal{P}_m.$$

We denote the cost of an extreme point p_m by:

$$c_m^{p_m} = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} stc_{km} z_{km}^t + \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} c_{km}^t r_{km}^t.$$

Therefore, the master problem can be formulated as follows.

$$z^{MDM} = \text{minimize} \quad \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} h_i s_i^t + \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} \bar{h}_{km} e_{km}^t \sum_{m \in \mathcal{M}} \sum_{p_m \in \mathcal{P}_m} c_m^{p_m} \bar{r}_m^{p_m}$$

$$+ \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}_{km}^t} cl_{km} \left(b_{km} - \sum_{i \in \mathcal{N}_k} l_i a_{ij} \right) y_{jkm}^t + \sum_{i \in \mathcal{N}} \tau_i s_i^T, \tag{21}$$

s.t. (7) and (15)

$$\sum_{p_m \in \mathcal{P}_m} r_{km}^{t\,p_m} \bar{r}_m^{p_m} + e_{km}^{t-1} - e_{km}^t = \sum_{j \in \mathcal{J}_{km}^t} y_{jkm}^t \quad \forall k \in \mathcal{K}, \ \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{22}$$

$$\textstyle\sum_{p_m \in \mathcal{P}_m} \bar{r}_m^{p_m} \le 1 \quad \forall m \in \mathcal{M}, \tag{23}$$

$$y_{jkm}^t, r_{km}^t, e_k^t, s_i^t \in \mathbb{Z}_+, \bar{r}_m^{p_m} \in \{0,1\} \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{J}_{km}^t, \forall p_m \in \mathcal{P}_m, \quad \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, \forall t \in \mathcal{T}. \tag{24}$$

The objective function (21) minimizes the inventory cost of items and objects, the cost of the production planning chosen for each machine, the waste cost and the inventory of items in the last period. Constraints (22) are the tactical demand constraints. The convexity constraints are given in (23). Constraints (24) define the domain of the variables.

To define the reduced cost of the subproblems associated with the production plans, consider the dual variables σ_{km}^t associated with constraints (22), and γ_m associated with constraints (23), $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$ and $\forall t \in \mathcal{T}$. The objective function of the subproblems consists of reducing the production cost and the setup cost over the extreme points. Thus, the subproblem related to machine m is given as follows.

$$\text{(Subproblem 2)} \quad rc2_m = \text{minimize} \quad \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} stc_{km} z_{km}^t + \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} (c_{km}^t - \sigma_{km}^t) r_{km}^t - \gamma_m, \tag{25}$$

$$\text{s.t.} \sum_{k \in \mathcal{K}} (st_{km} z_{km}^t + vt_{km} r_{km}^t) \le Cap_m \quad t \in \mathcal{T}, \tag{26}$$

$$r_{km}^t \le M_{km} z_{km}^t \quad k \in \mathcal{K}, t \in \mathcal{T}, \tag{27}$$

$$r_{km}^t \in \mathbb{Z}_+, z_{km}^t \in \{0,1\} \quad k \in \mathcal{K}, t \in \mathcal{T}. \tag{28}$$

The subproblems associated with the cutting-stock problem are Subproblems 1. In each iteration of the simplex method, Subproblems 1 and Subproblems 2 are solved. If $rc1_{km}^t \ge 0$ and $rc2_m \ge 0$ for all k, m and t, then we have an optimal solution for the linear relaxation of the master problem MDM.

The period decomposition model (PDM) can be equivalently described, where we have T subproblems and each one defines production plans for all machines in period t. Consequently, in this case there are $K \times M \times T + T$ subproblems to be solved in each iteration of the simplex method.

Both decompositions provide the same lower bound, which is equal or better than the lower bound provided by the linear relaxation of the pattern-oriented model. To show this fact, let $\underline{z}^{POM}$, $\underline{z}^{PDM}$ and $\underline{z}^{MDM}$ be the optimal solution of the linear relaxation of the pattern-oriented model, period decomposition model and machine decomposition model, respectively.

Proposition. $\underline{z}^{POM} \le \underline{z}^{PDM} = \underline{z}^{MDM}$.

Proof. See Appendix A. □

As described in Section 3.2, to find a feasible solution for MDM (and PDM), first the linear relaxation of the master problem is solved using the column generation method. Then, a feasible solution is obtained by a rounding heuristic applied to the resultant restricted master problem. In addition, we propose a heuristic for MDM that combines the column generation method and the adaptive large neighborhood search (CG–ALNS). The rounding heuristic used to obtain feasible solutions from the integer restricted master problem of the POM, PDM and MDM and the proposed heuristic for MDM are described in the following section.

4. Primal heuristics

In this section, we describe the two heuristics used in this paper to obtain feasible solutions for the 1CSP–CLSPP. The first one is a rounding heuristic that has shown to be efficient in the literature for some mathematical models. The second heuristic consists of combining column generation methods and the adaptive large neighborhood search to solve MDM.

4.1. Rounding heuristic

After solving the linear relaxation of the master problems by the classical column generation method, we apply a simple rounding heuristic to determine integer solutions. The heuristic consists of solving the integer restricted master problem with the columns used to optimally solve the linear relaxation by a MIP solver. However, in the integer restricted master problem, we consider only the basic columns of the cutting-stock problem where $y^t_{jkm} \geq 0.9$ in the linear optimal solution. This strategy can be efficient when a wide-ranging set of columns is available at the end of the column generation method. Otherwise, the generated columns cannot ensure a feasible solution for the original problem. We apply the heuristic after all three column generation methods implemented in this paper, i.e., the column generation methods to solve the linear relaxation of the POM, PDM and MDM.

4.2. Column generation heuristics and the adaptive large neighborhood search

We combine the column generation methods and the adaptive large neighborhood search (ALNS), which we denote by the CG–ALNS heuristic. MDM generates more diversified production plans than PDM, because production plan decisions are made for each machine considering all periods in the planning horizon. Therefore, we decided to perform the CG–ALNS method on MDM.

ALNS was introduced by Ropke and Pisinger [25] and is based on the large neighborhood search proposed by Shaw [26]. The method idea is to explore a wide diversity of neighborhoods by applying large changes in the incumbent solution and by accepting infeasible solutions. Several insertion and removal heuristics are used to perform such changes in the incumbent solution.

The standard framework of ALNS consists of applying an insertion heuristic and a removal heuristic in each iteration of the method. They are chosen according to a selection methodology, where the most used technique is the roulette wheel. This selection is based on the weight of each heuristic, which is adjusted at the end of a given number of iterations called segments. The weights are calculated according to the recorded scores of the heuristic success in obtaining new solutions. In general, the acceptance criteria of a solution is designed in a simulated annealing way.

This metaheuristic has shown to be efficient for solving different problems including the lot-sizing problem. An ALNS heuristic was proposed for the lot-sizing problem with setup times in [27]. The authors used a MIP solver as an operator of the method that solved a MIP subproblem. The mathematical model used by the authors in the hybrid ALNS heuristic is the classical one.

This section first describes a column generation heuristic. The idea of this routine is to provide an initial solution that may be feasible. Second, the ALNS layout is presented. Then, we provide the removal and insertion heuristics developed for the studied problem. The general layout of the column generation method combined with ALNS is described next.

4.2.1. Column generation heuristic

In order to generate more promising production plans and determine feasible solutions for the 1CSP–CLSPP, a heuristic based on the column generation method is performed (CG2 heuristic). The heuristic can be used not only to obtain initial solutions, but also iteratively with the ALNS (CG–ALNS heuristic).

As the usual column generation method, the CG2 heuristic is an iterative method where the linear relaxation of MDM and two types of subproblems are solved: Subproblem 1 and an adaptation of Subproblem 2 (see Section 3.3).

The heuristic starts with an initial feasible solution for the cutting stock problem constraints, i.e., a set of cutting patterns that can satisfy the demand of items, which provides the demand of objects of class k in each machine m and period t. To generate new production plans, the balancing constraints related to the production of objects are added to Subproblem 2. Therefore, for each $m \in \mathcal{M}$, the subproblem related to the lot-sizing problem is given by:

$$\text{(Subproblem 3)} \quad rc3_m = \text{minimize} \quad \sum_{k\in\mathcal{K}} \bar{h}_{km} e^t_{km} + \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} stc_{km} z^t_{km} + \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} c^t_{km} r^t_{km}, \tag{29}$$

$$\text{s.t. } r^t_{km} + e^{t-1}_{km} - e^t_{km} = \sum_{j \in \bar{\mathcal{N}}^t_{km}} \tilde{y}^t_{jkm} \quad \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \quad (26)\text{–}(28), \tag{30}$$

where $\tilde{y}^t_{jkm}$ is the number of objects of type k, produced in machine m cut according to pattern j needed in period t.

Before starting the CG2 heuristic, we consider two approaches to define the parameters $\tilde{y}^t_{jkm},\ j \in \bar{\mathcal{N}}^t_{km}, k \in \mathcal{K}, t \in \mathcal{T}, m \in \mathcal{M}$. The first one consists of solving the linear relaxation of MDM by the usual column generation. Then, the solution of variables y^t_{jkm} is rounded up. The second one is to use the cutting stock problem solution provided by the incumbent solution in the ALNS heuristic (see Section 4.2.2). In an attempt to generate more diversified production plans, the variables related to production plans determined in the usual column generation method are set to zero.

In each iteration of CG2 heuristic, M Subproblems 3 are solved. In this case, all new feasible production plans are added to the master problem. The linear relaxation of the restricted master problem MDM is solved and new dual variables are available to solve Subproblems 1. If profitable cutting patterns are provided ($rc1^t_{km} < 0$), they are added to the restricted master problem, and its linear relaxation is solved again. The method stops when no new feasible production plan is found or if $rc1^t_{km} \geq 0$ for all k, m, t. The CG2 heuristic is described in Algorithm 1.

Algorithm 1 CG2 heuristic.

Input: A set of cutting patterns that ensure a feasible solution for demand constraints (15) and parameters $\tilde{y}^t_{jkm},\ j \in \bar{\mathcal{N}}^t_{km}, k \in \mathcal{K}, t \in \mathcal{T}, m \in \mathcal{M}$;
Output: A set of solutions for 1CSP–CLSPP;

while Stop = False **do**
  Solve Subproblem 3, for all $m \in \mathcal{M}$;
  if at least a new feasible production plan has been found **then**
    Add the new feasible production plans to the restricted master problem MDM;
    Save the solution into the set of solutions;
  else
    Stop = True;
  end if
  Solve the linear relaxation of MDM and determine the dual variables;
  Solve Subproblem 1 for each $k \in \mathcal{K}, m \in \mathcal{M}, t \in \mathcal{T}$;
  if $rc1^t_{km} \geq 0, \forall k \in \mathcal{K}, m \in \mathcal{M}, t \in \mathcal{T}$ **then**
    Stop = True.
  else
    Add the cutting patterns with $rc1^t_{km} < 0$ to MDM;
    Solve the linear relaxation of MDM;
    Round up the value of the linear solution y^t_{jkm}, i.e., $\tilde{y}^t_{jkm} = \lceil \bar{y}^t_{jkm} \rceil$;
  end if
end while

4.2.2. Adaptive large neighborhood search

The proposed ALNS heuristic starts with an initial solution, a set of cutting patterns, a set of production plans and a lower bound. We implemented two insertion heuristics and four removal heuristics for the cutting stock problem, and three insertion heuristics and two removal heuristics for the lot-sizing problem. The removal and insertion heuristics are described in Sections 4.2.3 and 4.2.4, respectively. The master level of ALNS (solution acceptance and stop criteria) is based on simulated annealing.

In order to describe the ALNS framework, let $\mathcal{H}^-_C$ and $\mathcal{H}^+_C$ be the set of removal and insertion heuristics for the cutting stock problem, and $\mathcal{H}^-_L$ and $\mathcal{H}^+_L$ be the set of removal and insertion heuristics for the lot-sizing problem, respectively. The cutting stock removal heuristics remove a set of q_1 cutting patterns from the incumbent solution, where parameter q_1 defines the size of the neighborhood. Insertion heuristics try to insert different cutting patterns already known so that the demand of items is met. Similarly, the lot-sizing removal heuristic removes a set of objects produced by the machines, where the size of this set is defined according to parameter q_2. The lot-sizing insertion heuristics are used to complete the partial production plan obtained after the removal heuristic. Note that new production plans can be generated that are added to the restricted master problem MDM after performing the lot-sizing insertion heuristic.

The local search procedure is divided into segments with *seg* iterations. In each iteration of ALNS, one operator of each heuristic set $\mathcal{H}^-_L$, $\mathcal{H}^+_L$, $\mathcal{H}^-_C$ and $\mathcal{H}^+_C$ is chosen according to a roulette wheel selection. These selections depend on the weight of the heuristics. At the end of each segment j, the weight of the heuristics is updated and it has a value denoted by $w_{h,j}$, for each $h \in \mathcal{H} = \mathcal{H}^-_C \cup \mathcal{H}^+_C \cup \mathcal{H}^-_L \cup \mathcal{H}^+_L$. The weight of the heuristic in segment j depends on four parameters:

- δ_h: score of heuristic h obtained during the last segment;

- θ_h: number of times that heuristic h has been used in the last segment;
- rf: a reaction factor to control the effectiveness of the heuristic;
- $w_{h,j-1}$: the weight of heuristic h in the previous segment.

At the beginning of each segment, δ_h is set to zero. Then, in each iteration, the score can be increased by Δ_1, Δ_2 or Δ_3. In [25], Δ_1 is used if heuristic h resulted in a new global best solution. Δ_2 means that heuristic h resulted in a solution that has not been accepted before and its value is better than the value of the current solution. Δ_3 is considered if heuristic h resulted in a solution that has not been accepted before and its value is worse than the value of the current solution, but the solution was accepted.

To favor the variability of solutions, we consider a different strategy to increase the value of δ_h. In our method, we increase the score of heuristic h by Δ_1 if it provides a new and acceptable solution with a better value than the best solution. Δ_2 is used if heuristic h finds a new solution which is acceptable, but with a value worse than the best solution. Δ_3 is used for the heuristic that finds a solution that has not been accepted, but it is new. Preliminary computational tests showed that this strategy provided better solutions for 1CSP–CLSPP than the strategy described in [25].

Parameter θ_h is also set to zero at the beginning of each segment, and when a heuristic is used, its value is increased by one. The reaction factor rf controls how much the weight adjustment reacts to the score of the heuristic. Therefore, the weight of a heuristic h in a segment j is given by:

$$w_{h,j} = w_{h,j-1}(1 - rf) + rf\frac{\delta_h}{\theta_h}.$$

A heuristic $\bar{h} \in \mathcal{H}^*$ is selected with probability $w_{\bar{h}j}/\sum_{h\in\mathcal{H}^*} w_{h,j}$, for $\mathcal{H}^* = \mathcal{H}_L^-, \mathcal{H}_L^+, \mathcal{H}_C^-$ and $\mathcal{H}_C^+$.

As described in [25], the acceptance and the stop criteria follow the simulated annealing approach. The initial temperature in our algorithm is given by $T = T_0 * (s^* - LB)$, where s^* is the value of the initial solution (the best solution known so far), LB is the value of the optimal solution of the linear relaxation of MDM and T_0 is a fixed parameter. The temperature decreases in each iteration according to T*$cTemp$, where $0 < cTemp < 1$ is the cooling rate. After performing the four operators in the incumbent solution s_{ALNS} we obtain solution s', which is accepted with probability $\exp^{(f(s') - f(s_{ALNS}))/T}$. The pseudo-code of the ALNS algorithm that will embed in the proposed method for the 1CSP–CLSPP is shown in Algorithm 2.

4.2.3. ALNS removal heuristics

The removal heuristics address the disadvantages in a solution, except for the random operators. Parameters q_1 and q_2 control how many cutting patterns and machine productions will be removed from a solution, respectively. Given a solution $y^t_{jkm} = \tilde{y}^t_{jkm}$, the cutting stock removal heuristics sets variable y^t_{jkm} to a value smaller than $\tilde{y}^t_{jkm}$. The lot-sizing removal heuristics do not operate in the production plan variables in the master problem ($\bar{r}^{p_m}_m$), but in the quantity of the production of objects, i.e., variables r^t_{km} in the subproblem. This enables us to find new production plans. We use four removal heuristics in the cutting stock problem and two removal heuristics in the lot-sizing problem.

Cutting stock removal heuristics ($\mathcal{H}_c^-$)

Random cut removal. The random operator selects one cutting pattern in the current solution at random, where $\tilde{y}^t_{jkm} > 0$ and then reduces this value by one unit. This process is repeated q_1 times.

Worst cut removal. The operator removes from the solution q_1 cutting patterns with a higher trim loss from the solution.

Demand cut removal. This operator selects iteratively the item with the highest inventory in the last period T. Then, one unit of a cutting pattern that produces this item is randomly selected and removed. The maximum number of iterations performed in the demand cut removal is q_1.

Ahead cut removal. This operator is applied only to infeasible solutions. In each iteration, one negative inventory e^t_{km} is randomly selected. A negative inventory means that a object is cut without being produced. Then, a cutting pattern performed in this object type and period is randomly selected and removed. The operator performs q_1 iterations.

Lot-sizing removal heuristics ($\mathcal{H}_L^-$)

Random production removal. Given a production plan, we compute the total production of each machine m in each period t ($\sum_{k\in\mathcal{K}} r^t_{km}$ for each pair (m, t)). Then, the operator selects at random q_2 pairs (m, t) and removes the total production related to them completely.

Inventory production removal. The operator determines positive inventories in the last period T, i.e., $e^T_{km} > 0$. Among all pairs (m, t) such that machine m produces this type of object in period t, q_2 pairs (m, t) are selected and the total production related to them is completely removed.

4.2.4. ALNS insertion heuristics

In the removal heuristics, some cutting patterns and the production of some machines are removed and, consequently, some solutions can be infeasible. The infeasibility is characterized by a negative inventory stock of items or objects. Thus, ALNS insertion heuristics for the lot-sizing problem try to complete a partial solution and obtain a feasible solution, which can also define new production plans. In the cutting stock problem, the insertion heuristics try to include cutting patterns that are suitable to meet the residual demand.

Cutting stock insertion heuristics ($\mathcal{H}_C^+$)

Algorithm 2 ALNS.

Input: Best known solution s^*, lower bound LB, a set of cutting patterns, a set of production plans, reaction factor rf, q_1, q_2, seg, T_{min}, T_0;
Output: best known solution s^*, production plans;

$s_{ALNS} = s^*$;
$T = T_0 * (s^* - LB)$;
while $T \geq T_{min}$ **do**
  $\delta_h = 0$ and $\theta_h = 0$, for all $h \in \mathcal{H}$;
  for $i = 1$ to seg **do**
    $s' = s_{ALNS}$;
    Choose $h_C^- \in \mathcal{H}_C^-$ according to the roulette wheel to remove q_1 cutting patterns from s';
    Choose $h_C^+ \in \mathcal{H}_C^+$ according to the roulette wheel to insert the cutting patterns necessary into s' to satisfy the demand of items;
    Choose $h_L^- \in \mathcal{H}_L^-$ according to the roulette wheel to remove q_2 production plans from s';
    Choose $h_L^+ \in \mathcal{H}_L^+$ according to the roulette wheel to insert the production of objects and to generate new production plans in s';
    Increase θ_h for all chosen heuristics h;
    Add the feasible production plans to the restricted master problem MDM;
    if accept(s') **then**
      $s_{ALNS} = s'$;
      if $s' < s^*$ **then**
        Save s' in the set of Visited Solutions;
        $s^* = s'$; $\delta_h = \delta_h + \Delta_1$ for all heuristics chosen;
      else
        if s' is new **then**
          Save s' in the set of Visited Solutions;
          $\delta_h = \delta_h + \Delta_2$ for all heuristics chosen;
        end if
      end if
    else if s' is new and feasible **then**
      Save s' in the set of Visited Solutions;
      $\delta_h = \delta_h + \Delta_3$ for all heuristics chosen;
    end if
    $T = T * cTemp$;
  end for
  Update w_h for all insertion and removal heuristics.
end while

Greedy cut insertion. This operator consists of assigning a weight to all cutting patterns obtained by the method. The cutting patterns that can supply more items with a negative inventory and cause less cost in the objective function have more priority. Only cutting patterns that are at most 5% worse than the best one can be included in the solution. Among them, one cutting pattern is randomly chosen and added to the partial solution. The method performs this process iteratively until there is no negative inventory of items.

Idle capacity insertion. In this operator, each cutting pattern is evaluated and receives a value. The value of a cutting pattern j in period t is given by $(t + min\{-s_i^t, a_{ij}\}/100)$, where $s_i^t < 0$. Then, the cutting patterns are arranged in a non-increasing order of this value. One cutting pattern is selected at random and added to the solution. This process is performed iteratively while there is a negative inventory of items and idle capacity of at least one machine in a period up to t. In each iteration, the inventories, the idle capacity of the machines and the value of the cutting patterns are updated. At the end, if there is no idle capacity, but there is still a negative inventory of items, the greedy cut insertion operator is performed to complete the remaining demand.

Lot-sizing insertion heuristics $(\mathcal{H}_L^+)$

Random production insertion. This is an iterative operator, whereby each iteration determines all negative inventories e_{km}^t and selects one of them at random. Then, the object will be produced as much as the capacity of the machines allows between period 1 and t. This procedure is repeated a fixed number of iterations (it_{max}) or stops when there is no negative inventory.

Mixed integer programming production insertion. This operator tries to complete a partial solution of a production plan and, consequently, to obtain a new feasible production plan for machine m. Given a solution, we first determine the negative inventory stocks and update the capacity of the machines in each period t, obtaining the residual capacity of the machines ($RCap^t_m$). Then, Subproblems 3 are solved, where $\sum_{j\in\bar{\mathcal{N}}^t_{km}} \tilde{y}^t_{jkm} = \begin{cases} -e^t_{km}, & \text{if } e^t_{km} < 0 \\ 0, & \text{otherwise} \end{cases}$ and Cap_m is replaced by $RCap^t_m$.

Dynamic production insertion. This operator follows the same idea of the mixed integer programming production insertion operator. However, instead of solving Subproblems 3, we solve the following knapsack problem with setups for each idle machine m in period t, for $t = T, \ldots, 1$ and $m = 1, \ldots, M$.

$$\text{maximize} \quad \sum_{k\in\mathcal{K}} r^t_{km} - \sum_{k\in\mathcal{K}} stc_{km} z^t_{km}, \tag{31}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} (st_{km} z^t_{km} + vt_{km} r^t_{km}) \le RCap^t_m, \tag{32}$$

$$0 \le r^t_{km} \le \max\{0, -e^t_{km} z^t_{km}\}, \quad \forall k \in \mathcal{K}, \tag{33}$$

$$r^t_{km} \in \mathbb{Z}_+, z^t_{km} \in \{0, 1\} \quad k \in \mathcal{K}. \tag{34}$$

where $RCap^t_m$ is the residual capacity of the machines obtained after performing a lot-sizing removal heuristic. The objective function (31) maximizes the total number of objects produced to satisfy the demand of objects and minimizes the setup times. The demand is given by the negative inventory stocks determined according to the partial solution. Subproblems (31)–(34) were solved by the dynamic programming described in [28].

4.2.5. Combining column generation methods and ALNS

The column generation methods cooperate with the ALNS heuristic by providing cutting patterns, production plans and complete solutions. First, the linear relaxation of MDM is solved by the column generation method. After solving the model, a set of cutting patterns and production plans are available and are introduced in the column generation heuristic (CG2) described in Algorithm 1. Then, the ALNS and CG2 heuristics are solved iteratively, where CG2 can provide new solutions, cutting patterns and production plans for ALNS. The method is performed a fixed number of iterations (*ite*). At the end of the method, the integer restricted master problem MDM is solved by a MIP solver. This model keeps all cutting patterns and the production plans generated by both column generation methods and the production plans generated by the ALNS heuristic. Moreover, the best known solution is introduced to the MIP solver. Algorithm 3 shows the combined method

Algorithm 3 CG–ALNS.

```
Solve linear relaxation of MDM;
Run Algorithm 1;
for i = 1 to ite do
   Run Algorithm 2;
   Run Algorithm 1;
end for
Solve the integer restricted master problem MDM by the MIP solver with the best known solution introduced as a MIP
start solution.
```

which we denote by CG–ALNS.

5. Computational results

In this section, we report our computational experience of the item-oriented model, column generation methods, rounding heuristic and CG–ALNS heuristic. The tests were performed on an Intel Core i7 at 3.20 GHz and 16 GB RAM. The methods were implemented in language C/C++. To solve the knapsack problems that determine the cutting patterns, we implemented a branch-and-bound method. The master problems, Subproblems 1, Subproblems 2, and IOM were solved by CPLEX 12.6. We carried out computational experiments using two sets of instances. The first one (Set 1) was proposed by Poltroniere et al. [2]. For the second set of instances (Set 2), we adapted the real-world data of Santos and Almada-Lobo [5] that were collected from a paper mill company.

The remainder of this section is organized as follows. Section 5.1 describes the parameter settings for the methods. In Section 5.2, we described how the parameters of CG–ALNS were tuned. The instances from Set 1 and their computational results are reported in Section 5.3. Set 2 and its computational results are given in Section 5.4.

5.1. Parameter settings

The computational time to solve the IOM by CPLEX was limited to 1800 s. The upper bound on the number of objects O^t_{km} needed to satisfy the demand is given by the maximum number of objects of grade k that can be produced in machine m in period t. The linear relaxation of the master problems was solved by the column generation method. When an optimal

Table 1
Parameters experimented during the tuning phase.

ite	1, **2**, 3
cTemp	0.995, **0.997**, 0.999
$\bar{q}_1$	0.10, **0.20**, 0.30
$\bar{q}_2$	0.10, **0.20**, 0.30
rf	0.3, **0.5**, 0.7
$(\Delta_1, \Delta_2, \Delta_3)$	**(50, 15, 30)**, (50, 15, 20), (50,30,30), (50, 40, 30), (40, 15, 30), (60, 15, 30)

solution was found, the integer restricted master problem was solved by CPLEX, i.e., we applied the rounding heuristic described in Section 4.1. To solve the integer restricted master problem, the computational time was limited to 600s. In addition, we set the optimality gap tolerance to 0.2%.

For the CG–ALNS heuristic, the optimality gap tolerance for Subproblems 3 in the CG2 heuristic was also set to 0.2% and the computational time was limited to 120 s. In the CG–ALNS heuristic, first the linear relaxation of MDM is solved to optimality (line 1 in Algorithm 3). After that, lines 2–7 in Algorithm 3 must be performed at a maximum of 600 s, where Algorithms 1 and 2 were run with a time limit of 510 s. Then, the integer restricted master problem (line 7) was solved with a computational time limited to the maximum between 90 s and the remaining time to complete 600 s. The number of segments in Algorithm 2 (*seg*) was fixed to 100 iterations, $T_0 = 0.035$, $T_{min} = 1$ and the maximum number of iterations in the random production insertion heuristic is $it_{max} = 100$.

5.2. Tuning parameters of CG–ALNS

To tune the parameters of CG–ALNS, we used seven instances that were challenging to solve in preliminary computational tests. The instances used are six instances from Set 2 (7, 11, 12, 13, 15 and 16) and one instance in class 22 from Set 1. The parameters that must be fixed are: *ite, cTemp, rf,* Δ_1, Δ_2, Δ_3, q_1 and q_2. *ite* is the total number of iterations of CG–ALNS. *cTemp* is the cooling rate of Simulated Annealing. *rf,* Δ_1, Δ_2, and Δ_3 are used to adjust the weight of the heuristics. q_1 is the coefficient that determines the scope of the cutting stock removal operators, which is given by $q_1 = \lfloor \bar{q}_1 * |\overline{\mathcal{J}}^t_{km}| \rfloor$, where $0 \le \bar{q}_1 \le 1$ and $\overline{\mathcal{J}}^t_{km} \subset \mathcal{J}^t_{km}$ is the set of basic cutting patterns defined after solving the linear relaxation of MDM (see Section 3.2). In addition, q_2 is the coefficient that determines the scope of the lot-sizing removal operators, where $q_2 = \lfloor \bar{q}_2 * M * T \rfloor$, for $0 \le \bar{q}_2 \le 1$.

The first set of parameters was defined while developing the heuristic. Then, in each turn of the tuning phase, one parameter took several values while the remaining ones were kept fixed. In each parameter configuration, the heuristic was run five times, each one with a different random seed. To define the best value for the parameters, we considered the average results. The parameters used in the configuration phase are given in Table 1. The best average results were obtained using $ite = 2$, $cTemp = 0.997$, $\bar{q}_1 = 0.20$, $\bar{q}_2 = 0.20$, $rf = 0.5$ and $(\Delta_1, \Delta_2, \Delta_3) = (50, 15, 30)$.

5.3. Computational results of Set 1

5.3.1. Instances description

The instances of Poltroniere et al. [2] are classified into 27 classes by varying the number of periods (T), object grade types (K) and item types for each grade type ($|\mathcal{N}_k|$). There are two machines to produce the items ($M = 2$), where one machine produces objects of length $b_{k1} = 540$ cm, and the other one produces objects of $b_{k2} = 460$ cm. For each item, the length was generated randomly such that $l_i \in [0.1, 0.3]. \sum_{m \in \mathcal{M}} b_{km}/M$ for all $k \in \mathcal{K}$, and the demand is given by $d_i^t \in [0, 300]$ for all $t \in \mathcal{T}$. If $d_i^t < 50$ then $d_i^t = 0$. For each class, there are 10 instances. Further details on the instances can be found in Poltroniere et al. [2].

5.3.2. Lower bounds

In this section, we evaluate the performance of the column generation methods to solve the linear relaxation of the POM, PDM and MDM, as well as the performance of CPLEX to solve the linear relaxation of the IOM. The measurements used to compare the methods are the average time to solve the linear relaxation, the number of iterations necessary to obtain an optimal solution and the average gap. The gap is given by $100 * (\underline{z}^{DW} - \underline{z}^{POM})/\underline{z}^{DW}$, where $\underline{z}^{\,POM}$ is the lower bound of the pattern-oriented model and $\underline{z}^{\,DW}$ is the lower bound of PDM or MDM, that provide the same lower bound. The gap in the linear relaxation of IOM is given by $100 * (\underline{z}^{IOM} - \underline{z}^{POM})/\underline{z}^{POM}$.

Three instances out of 270 were proved by the methods to be infeasible, each one in classes 4, 7 and 16, respectively. The computational time to solve the POM is always smaller than the computational time to solve the period/machine decomposition models, where the average computational time to solve the POM is 4 s. In addition, the average computational time to solve the MDM is smaller than the one to solve the PDM (81 and 93 s, respectively). However, the number of iterations of the column generation method to solve MDM is always higher. The average number of iterations of the method is 23, 25 and 93 when performed on the POM, PDM and MDM, respectively. Solving the subproblems by CPLEX is reasonably fast, however, the time is increased by writing and sending the subproblems to CPLEX and saving the optimal solution. As expected, there was not enough memory available to store IOM for large-sized test problems on the computer (classes 12, 15, 18, 21, 24 and 27). Moreover, the average running time of CPLEX on the solved instances was higher than the column generation methods, where the average running time of CPLEX to solve the IOM is 188 s on average.

The gap between PDM/MDM and POM is very similar (0.004% on average). The lower bound provided by the linear relaxation of IOM is worse than the remaining models (2.17% worse than the POM on average), except for class 6, where the lower bound provided by IOM is equal to POM. Although the computational time to solve the linear relaxation of PDM and MDM is higher than the column generation method when solving the POM, their lower bound has shown to be slightly better than the lower bound provided by the POM. This fact can be useful when the integer master problem is solved to optimality as computational experiments have shown that CPLEX has some difficulty in solving the integer restricted master problems to optimality.

5.3.3. Feasible solutions

In this section, we evaluate the performance of CPLEX to solve the IOM, the rounding heuristic applied to the POM, PDM and MDM, and the CG–ALNS heuristic. For CG–ALNS, the outcome of five runs with different random seeds is analyzed. For each solution approach, we analyzed: i) the number of instances solved; ii) the gap; iii) the computational solution time; iv) the waste material; and v) the inventory of final items at the end of period T. The gap is given by $100 * (z_{IP} - \underline{z})/z_{IP}$, where z_{IP} is the value of the solution obtained by the methods and $\underline{z}$ is the value of its linear relaxation obtained by the column generation method. For the CG–ALNS, we considered the best, worst and the average results of each class of five runs.

CPLEX did not find a feasible solution for classes 10–27 for the PDM, MDM and IOM. Furthermore, for the IOM no feasible solution was found for classes 3, 5, 6, 8, 9. In addition, CPLEX found 76 feasible solutions in the PDM, 81 in the MDM, and 14 in the IOM out of 90 instances (Classes 1–9). The computational running time, the gap and the material waste are higher than those obtained in RH applied to the POM and in the CG–ALNS, mainly on account of the MDM and IOM. The IOM reached the time limit with an average GAP of 42.82%, while the average computational time and GAP of the PDM are given by 285 s and 7.85%, and MDM are given by 154 s and 20.81%, respectively. The average waste material and inventory of items at the end of period T in PDM are 0.47% and 0.08%, while in MDM they are 1.81% and 0.04% and IOM are 6.95% and 0.18%, respectively.

In RH applied to the POM, CPLEX was able to solve 265 instances. It did not find feasible solution for two instances (one instance in class 15 and one in class 26). For one instance in classes 4, 7 and 16, the linear relaxation of the master problem is infeasible (See Section 5.3.2). CPLEX was able to determine a solution to the POM restricted master problem within the given optimality gap (0.2% of tolerance) in 600 s for 60% of the instances. However, proving the optimality on the integer restricted master problem is a time-consuming process. The average optimality gap of RH applied to the POM is 0.31%. If we consider the gap of each instance, the largest optimality gap is 0.97%.

The CG–ALNS was able to solve 266 instances, which is one instance more than RH applied to the POM. The solutions obtained by the CG–ALNS are very close to the solutions obtained by RH applied to the POM. For these instances, the average gap obtained by CG–ALNS is 0.13% worse than RH applied to the POM. However, for classes 12, 15, 21, 24 and 27, its average gap was better. The computational time of CG–ALNS is higher since the computational time of the column generation to solve the linear relaxation of MDM is higher than the column generation method to solve POM. In addition, after obtaining the optimal solution for the linear relaxation of MDM, the computational time of CG–ALNS is limited to 600 s.

The material waste and inventory of items at the end of period T are very small in both methods. In the RH performed on the POM, the average percentages of waste material and inventory of items at the end of period T are 0.24% and 0.02%, respectively. The average waste material and inventory of items at the end of period T obtained by the CG–ALNS heuristic are 0.07% and 0.00% (the average of the best scenarios), 0.26% and 0.02% (the average results), 0.71% and 0.02% (the average of the worst scenarios), respectively.

Before running line 7 of Algorithm 3, in all 266 instances solved by CG–ALNS, there was a feasible solution available. This intermediate solution was added to CPLEX as a MIP start solution (line 7 of Algorithm 3). The average gap of these intermediate solutions is 0.63%, which was obtained in an average running time of 162 s.

We do not compare the objective values obtained from our methods with Poltroniere et al. [2,3], because the feasible regions can be different and the objective function of our models has an additional term to minimize the inventory of items at the end of period T. Poltroniere et al. [2,3] considered a set of additional constraints that uses an estimation of the weight of waste occurred in the cutting process. This value is calculated in each iteration of the heuristics in [2] and in each iteration of the column generation method applied to the linear relaxation of the model in [3]. In addition, the RH applied to the POM and the CG–ALNS solved 17 and 18 more instances than best Lagrangian heuristic of Poltroniere et al. [2], and 24 and 25 more instances then the column generation heuristic of Poltroniere et al. [3], respectively.

CPLEX is very efficient in terms of solving the classical model of the lot-sizing problem, and the same is true for the integrated problem. Therefore, the rounding heuristic applied to the POM provides high quality solutions for the instances of Poltroniere et al. [2]. However, the rounding heuristic applied to the PDM and MDM is not efficient due to the poor variety of production plans available at the end of the column generation method. This problem was overcome by the CG–ALNS heuristic that was applied to the MDM, which shows to be competitive and provided a better solution than the rounding heuristic performed on the POM for some instances of Poltroniere et al. [2].

Table 2
Computational results of the CG–ALNS heuristic and RH applied to POM to obtain upper bounds in Set 2.

Inst.	K/T/n	CG–ALNS heuristic			Time (s)	RH performed on POM	
		GAP					
		Best	Worst	Av.		Gap (%)	Time (s)
1	1/3/34	12.84	12.84	12.84	602.64	25.61	602.20
2	2/3/41	15.84	15.84	15.84	603.34	23.94	4.15
3	3/3/52	11.70	11.70	11.70	612.29	14.43	5.72
4	4/4/57	10.43	10.45	10.45	614.80	12.40	602.10
5	5/4/69	11.54	11.56	11.55	627.67	14.72	3.88
6	6/4/71	11.81	11.85	11.83	618.82	14.82	4.15
7	7/6/100	9.29	9.78	9.41	641.98	11.04	606.47
8	8/7/113	7.89	9.42	8.82	684.10	8.95	606.11
9	9/7/116	8.28	9.04	8.52	731.76	9.05	607.76
10	10/8/129	7.52	8.23	7.94	811.08	9.12	609.26
11	11/8/146	9.38	10.45	9.64	811.85	10.31	614.74
12	12/9/155	8.58	9.63	9.10	936.91	9.90	614.30
13	13/9/165	9.73	10.57	10.19	1105.64	10.94	616.75
14	14/10/173	10.37	11.32	10.66	1354.67	12.03	618.21
15	15/11/186	10.84	12.00	11.31	1666.37	11.89	619.77
16	16/11/189	10.85	11.64	11.31	1804.78	12.26	620.41
Av.		10.43	11.02	10.69	889.29	13.21	459.75

5.4. Computational results of Set 2

This dataset was used in Santos and Almada-Lobo [5]. In the production unit studied, there is one paper machine available to produce the objects. We generated 16 instances by varying the number of grades from 1 to 16, where each grade has up to 34 types of required items. The planning horizon has 20 periods. However, we consider that the number of periods active in the planning horizon depends on the number of grades in the instance. For example, instance number 1 has only one grade type and only three periods are required to produce all the demands. For the instances with 16 grades, 11 periods are sufficient to meet the demand. To advance the production, we set the machine capacity to zero in the remaining periods. The real production process avoids shutdowns of the paper machine. Therefore, producing the demand at the beginning of the planning horizon can have two advantages. Firstly, it shows the capacity available for production where new demands can be accepted. Secondly, it provides a tighter production sequence of grades to reduce equipment down-time.

The industry considers the customer demand in tons. However, the demand can be estimated in units. For this, we consider that the paper is rolled in an internal cardboard tube of 10cm diameter with a paper density of 1.61 g/cm^3. To ensure that the demands of all customers are satisfied, non-integer required items were rounded up.

To determine some objective function costs, we calculated the average selling price of the items grouped by grade type and width. An item inventory cost is defined as 0.5% of its average price per year, which corresponds to 1/36,500 of the object's selling price per day. The object inventory cost is defined as 1% of the average annual price of the jumbo. The production cost of the object is also 1% of its average price. The paper waste cost is given by the object price divided by its length. For example, an object of 2000 mm of a specific grade sold for $800 corresponds to the paper waste cost of $800/2000 mm = $0.40/mm.

When the machine changes production from one type of paper to another, the paper of intermediate grade is recycled, where 20% of the paper value accounts for recycling costs. Then, we considered the setup cost for each paper as 20% of the paper value times the sale price. The calculus of the setup time is similar to the setup cost. The setup time and cost are sequence dependent, but we use the average value of them.

The production time for each object type is given by the production rate of the paper machine in tons/hour and the size of the object. The size of the object is defined according to the largest diameter of the items. Columns 1 and 2 in Table 2 show some details of Set 2. The first column (Inst.) represents the instance number, column K/T/n shows the number of grades, the number of periods active and the total number of required items, respectively.

We also analyze the performance of the column generation method to solve the linear relaxation of the POM, PDM and MDM, and the results of the linear relaxation of the IOM solved by CPLEX on instances of Set 2. The gap of MDM and PDM is given by: $100 * (\underline{z}^{DW} - \underline{z}^{POM})/\underline{z}^{DW}$, where $\underline{z}^{DW}$ is the lower bound of PDM or MDM. The gap of IOM is given by $100 * (\underline{z}^{IOM} - \underline{z}^{POM})/\underline{z}^{POM}$. The running time of the column generation method performed on the linear relaxation of POM is considerably smaller than when it is performed on the PDM and MDM. Moreover, the computational time to solve the linear relaxation of the MDM is higher than the PDM for large instances. The average running time of the column generation method performed on the POM, PDM and MDM is 7 s, 38 s and 287 s. The average number of iterations of the column generation performed on the POM, PDM and MDM is 47, 53 and 475, respectively. The lower bound provided by the PMD and MDM is stronger, where the average GAP of MDM\PDM is 0.10%. In addition, the deviation between the linear relaxation of the POM and PDM\MDM for these instances is higher than the instances of Set 1. The linear relaxation of the IOM results in the worst lower bound, which is 79.01% worse on average than the POM.

Table 2 shows the average gap and the running time of the CG–ALNS heuristic and rounding heuristic applied to POM. The optimality gap is given by $100 * (z_{IP} - \underline{z})/z_{IP}$, where z_{IP} is the integer solution found by the heuristics, and $\underline{z}$ is the optimal value of the relaxed master problem. Columns 3–5 show the best, worst and average gaps obtained by the CG–ALNS heuristic in five runs. The average computational time of the CG–ALNS is shown in column 6. The gap and running time of the rounding heuristic applied to the POM are shown in columns 7 and 8.

The CG–ALNS can provide better solutions for all instances of Set 2. The average solution value provided by CG–ALNS is better than the solution value obtained by RH applied to the POM for all instances. In the worst scenario, the CG–ALNS obtained a worse solution than RH applied to the POM for four instances: 8, 9, 11 and 15. Although the worst solution obtained by CG–ALNS in instance 9 is worse than the solution obtained by RH applied to the POM, the gap is better than the gap obtained by the RH applied to the POM. After solving the linear relaxation of the POM and MDM, the computational time of the CG–ALNS and rounding heuristic was limited to 600 s. Since the computational time necessary to solve the linear relaxation of MDM was considerably higher than the linear relaxation of the POM, the total running time of CG–ALNS was higher than the total running time of RH applied to POM. The last line in Table 2 shows the average results of Set 2 considering all 80 runs (5 runs for 16 instances).

The material waste obtained by CG–ALNS is slightly higher than RH applied to POM, where the average values are 3.29% and 3.22%, respectively. However, the percentage of the inventory of items in period T is smaller in CG–ALNS, where the average values are 3.50% in CG–ALNS and 3.75% in RH applied to POM. The worst and the best material waste and inventory of items at the end the period T in CG–ALNS are very close to the average results.

The instances of Set 2 are more difficult to solve than instances of Set 1. However, computational results show that the rounding heuristic performed on the POM and the CG–ALNS heuristic are capable of obtaining good solutions for these instances. According to the industry, the percentage of material waste is about 3%, which is equivalent to the results obtained by CG–ALNS and RH performed on the POM. The percentage of material waste can decrease if we anticipate the production of some items with high demands. In practice, the industry allows backordering due to the limited manufacturing capacity in contrast to large orders constantly received. As a result, the industry cuts some items that have high demands in order to avoid material waste. However, we do not have information about other items with high demands that could be included in the data set.

Before running line 7 of Algorithm 3, a feasible solution was available for all instances in the five runs. For instances 1, 2 and 3, this intermediate solution is better than the solution provided by RH performed on POM. The average gap of these intermediate solutions is 14.75%. The intermediate solution was added to CPLEX as a MIP start solution (line 7 of Algorithm 3), which was quickly improved by CPLEX.

The rounding heuristic performed on PDM and MDM obtained a feasible solution only for Instance 1 with gap of 35.56% and 35.35%, respectively, which is worse than the solution obtained by POM. The computational time to solve these instances was smaller than 13 s in both methods. In addition, CPLEX found feasible solutions on IOM for instances 1, 2 and 4. The CPLEX gap for instance 1 is 59.63%, for instance 2 is 39.81%, and for instance 4 is 83.27%.

We can conclude that the rounding heuristic performed on POM and CG–ALNS provided good solutions for instances in Set 2. While our computational results look favorable, improvements may still be possible in terms of material waste minimization by considering forecast demands.

6. Conclusions

We investigated three mathematical formulations for the one-dimensional cutting-stock problem integrated with the lot-sizing problem: the pattern-oriented model, the period decomposition model and the machine decomposition model. The column generation method was used to solve the linear relaxation of the models. The lower bound provided by the period and machine decomposition models was slightly better than the lower bound of the pattern-oriented model. After solving the relaxed master problem, the integer restricted master problem was solved using a MIP solver (Rounding heuristic). In addition, we proposed a heuristic that combines the column generation method with the adaptive large neighborhood search (CG–ALNS) to solve the machine decomposition model.

The column generation methods, the rounding heuristics and CG–ALNS were evaluated in instances taken from the literature and instances based on a real-world application. Computational results show that CPLEX 12.6 solver applied to the restricted master problem of the pattern-oriented model provides almost optimal solutions for instances of the literature. However, for instances adapted from real-world data the gap is considerably high. Due to the poor variety of production plans in the restricted master problem of period and machine decomposition models obtained after the column generation method, the rounding heuristic cannot provide feasible solution for most instances. However, this difficulty was overcome using the CG–ALNS heuristic applied to solving the machine decomposition model. The CG–ALNS shows competitive results for instances from the literature and provided better solutions for all instances adapted from real-world data.

Our future work will be to develop a branch-and-price method for the machine decomposition model. We will use the CG–ALNS heuristic to improve the performance of the branch-and-price method. Other research direction could be to consider other problems that occur in paper industries, such as sequence-dependent setups in the production of objects and production of cellulose in the digester.

Acknowledgments

The authors would like to thank the anonymous referee and the associate editor for their constructive comments, which have significantly improved the presentation of this paper. This research was supported by FAPESP (2008/09046-5 and 2012/21176-7) and CNPq (306918/2014-5).

Appendix A. Proof of proposition

Proposition. $\underline{z}^{POM} \leq \underline{z}^{PDM} = \underline{z}^{MDM}$.

Proof. The inequality follows from the fact that applying the Dantzig–Wolfe decomposition, the lower bound of the decomposed problem can never be smaller than the one provided by the pattern-oriented model.

First of all, let us focus only on the columns obtained by the subproblems (26)–(28) of the machine decomposition. The term that represents the value of these columns in the objective function (21) is given by:

$$\sum_{m\in\mathcal{M}}\sum_{p_m\in\mathcal{P}_m} c_m^{p_m}\bar{r}_m^{p_m}.$$

This is the only term that is different from the master problem objective function of the period decomposition, which is written as follows:

$$\sum_{t\in\mathcal{T}}\sum_{p_t\in\mathcal{P}_t} c_t^{p_t}\bar{r}_t^{p_t},$$

where $c_t^{p_t} = \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} stc_{km}z_{km}^t + \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} c_{km}^t r_{km}^t$.

To prove the equality, consider the optimal solution $\underline{z}^{MDM}$ obtained by the machine decomposition model. For each subproblem m in (26)–(28), $m = 1, \ldots, M$, we represent the solution of a production plan p_m in the following matrix.

$$\begin{bmatrix} r_{1m}^1 & \cdots & r_{Km}^1 & z_{1m}^1 & \cdots & z_{Km}^1 \\ r_{1m}^2 & \cdots & r_{Km}^2 & z_{1m}^2 & \cdots & z_{Km}^2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{1m}^T & \cdots & r_{Km}^T & z_{1m}^T & \cdots & z_{Km}^T \end{bmatrix}, \tag{35}$$

where each line t $(t = 1, \ldots, T)$ of the matrix represents a feasible constraint in (26), i.e., $\sum_{k\in\mathcal{K}}(st_{km}z_{km}^t + vt_{km}r_{km}^t) \leq Cap_m$ for each $t = 1, \ldots, T$. The value of this solution in the master problem is given by:

$$\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}(stc_{km}z_{km}^t + c_{km}^t r_{km}^t)\bar{r}_m^{p_m}. \tag{36}$$

By choosing line t of a production plan p_m, $m = 1, \ldots, M$, we can compose a new matrix, that is given by:

$$\begin{bmatrix} r_{11}^t & \cdots & r_{K1}^t & z_{11}^t & \cdots & z_{K1}^t \\ r_{12}^t & \cdots & r_{K2}^t & z_{12}^t & \cdots & z_{K2}^t \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{1M}^t & \cdots & r_{KM}^t & z_{1M}^t & \cdots & z_{KM}^t \end{bmatrix}. \tag{37}$$

Note that:

- Matrix (37) represents a feasible solution of the t-th subproblem in the period decomposition, i.e., $\sum_{k\in\mathcal{K}}(st_{km}z_{km}^t + vt_{km}r_{km}^t) \leq Cap_m$ for $m = 1, \ldots, M$. Thus, it represents a production plan p_t.
- By choosing line t of a subset of subproblems and completing some lines of (37) with zero, it is possible to obtain more than one matrix of type (37). These matrices represent a subset of production plans $\tilde{\mathcal{P}}_t \subset \mathcal{P}_t$.
- The arrangement of these production plans p_t must be done for all $t \in \mathcal{T}$ so that the sum of their values must be equal to (36), i.e.:

$$\sum_{t\in\mathcal{T}}\sum_{p_t\in\tilde{\mathcal{P}}_t}(stc_{km}z_{km}^t + c_{km}^t r_{km}^t)\bar{r}_t^{p_t} = \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}(stc_{km}z_{km}^t + c_{km}^t r_{km}^t)\bar{r}_m^{p_m}. \tag{38}$$

We have shown that a feasible solution of the machine decomposition model is also feasible for the period decomposition model. Then, $\underline{z}^{PDM} \leq \underline{z}^{MDM}$. Equally, we can show that $\underline{z}^{MDM} \leq \underline{z}^{PDM}$, which follows the equality. $\square$

References

[1] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting-stock problem, Oper. Res. 9 (6) (1961) 849–859.
[2] S.C. Poltroniere, K.C. Poldi, F.M.B. Toledo, M.N. Arenales, A coupling cutting stock-lot sizing problem in the paper industry, Ann. Oper. Res. 157 (1) (2008) 91–104.
[3] S.C. Poltroniere, S.A. Araujo, K.C. Poldi, Optimization of an integrated lot sizing and cutting stock problem in the paper industry, Tend. Mat. Apl. Comput. 17 (3) (2016) 305–320.
[4] L.V. Kantorovich, Mathematical methods of organizing and planning production, Manag. Sci. 6 (4) (1960) 366–422.
[5] M.O. Santos, B. Almada-Lobo, Integrated pulp and paper mill planning and scheduling, Comput. Ind. Eng. 63 (1) (2012) 1–12.
[6] E.V. Krichagina, R. Rubio, M.I. Taksar, L.M. Wein, A dynamic stochastic stock-cutting problem, Oper. Res. 46 (5) (1998) 690–701.
[7] P. Keskinocak, F. Wu, R. Goodwin, S. Murthy, R. Akkiraju, S. Kumaran, et al., Scheduling solutions for the paper industry, Oper. Res. 50 (2) (2002) 249–259.
[8] S. Menon, L. Schrage, Order allocation for stock cutting in the paper industry, Oper. Res. 50 (2) (2002) 324–332.
[9] A. Respicio, M. Captivo, A. Rodrigues, A dss for production planning and scheduling in the paper industry, in: International Conference on Decision Making and Decision Support in the Internet Age, University College Cork, Cork, 2002, pp. 298–308.
[10] M. Correia, J.F. Oliveira, J. Ferreira, Reel and sheet cutting at a paper mill, Comput. Oper. Res. 31 (8) (2004) 1223–1243.
[11] J.M. Valério de Carvalho, Exact solution of bin-packing problems using column generation and branch-and-bound, Ann. Oper. Res. 86 (2012) 629–659.
[12] M. Correia, J.F. Oliveira, J.S. Ferreira, Integrated resolution of assignment, sequencing and cutting problems in paper production planning, Int. J. Prod. Res. 50 (18) (2012) 5195–5212.
[13] K. Kim, B.I. Kim, H. Cho, Multiple-choice knapsack-based heuristic algorithm for the two-stage two-dimensional cutting stock problem in the paper industry, Int. J. Prod. Res. 52 (19) (2014) 5675–5689.
[14] M.C.N. Gramani, P.M. França, The combined cutting stock and lot-sizing problem in industrial processes, Eur. J. Oper. Res. 174 (1) (2006) 509–521.
[15] M. Gramani, P. França, M. Arenales, A Lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem, Int. J. Prod. Econ. 119 (2) (2009) 219–227.
[16] M.C.N. Gramani, P.M. França, M.N. Arenales, A linear optimization approach to the combined production planning model, J. Frankl. Inst. 348 (7) (2011) 1523–1536.
[17] E. Silva, F. Alvelos, C.J.M. Valerio, Integrating two-dimensional cutting stock and lot-sizing problems, J. Oper. Res. Soc. 65 (1) (2014) 108–123.
[18] A. Farley, Mathematical-programming models for cutting-stock problems in the clothing industry, J. Oper. Res. Soc. 39 (1) (1988) 41–53.
[19] L. Hendry, K. Fok, K. Shek, A cutting stock and scheduling problem in the copper industry, J. Oper. Res. Soc. 47 (1) (1996) 38–47.
[20] M. Reinders, Cutting stock optimization and integral production planning for centralized wood processing, Math. Comput. Model. 16 (1) (1992) 37–55.
[21] S.L. Nonås, A. Thorstenson, A combined cutting-stock and lot-sizing problem, Eur. J. Oper. Res. 120 (2) (2000) 327–342.
[22] S.L. Nonås, A. Thorstenson, Solving a combined cutting-stock and lot-sizing problem with a column generating procedure, Comput. Oper. Res. 35 (10) (2008) 3371–3392.
[23] C. Arbib, F. Marinelli, Integrating process optimization and inventory planning in cutting-stock with skiving option: An optimization model and its application, Eur. J. Oper. Res. 163 (3) (2005) 617–630.
[24] P. Vance, Branch-and-price algorithms for the one-dimensional cutting stock problem, Comput. Optim. Appl. 9 (3) (1998) 211–228. Cutting stock problems. In: Column Generation; chap. 5. Springer Science + Business Media, 2005; 2005, p. 131–162
[25] S. Ropke, D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, Transp. Sci. 40 (4) (2006) 455–472.
[26] P. Shaw, A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems, Technical report, Department of Computer Science, University of Strathclyde, Scotland, 1997.
[27] L.F. Muller, S. Spoorendonk, D. Pisinger, A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times, Eur. J. Oper. Res. 218 (3) (2012) 614–623.
[28] S. Michel, N. Perrot, F. Vanderbeck, Knapsack problems with setups, Technical report 00232782, INRIA, 2008. Available at https://hal.inria.fr/inria-00232782v1.