TOP (2020) 28:178–192
https://doi.org/10.1007/s11750-019-00531-3

ORIGINAL PAPER

A matheuristic algorithm for the one-dimensional cutting stock and scheduling problem with heterogeneous orders

Anselmo Ramalho Pitombeira-Neto[1] · Bruno de Athayde Prata[1]

Received: 18 February 2019 / Accepted: 9 September 2019 / Published online: 16 September 2019

Abstract
Cutting stock and bin packing problems are common in several industrial sectors, such as paper, glass, furniture, steel industry, construction, transportation, among others. The classical cutting stock problem (CSP) ignores the production planning and scheduling of multiple customer orders. Nevertheless, in real industrial settings customer orders have to be planned and scheduled over time so as to meet demand and required due dates. We propose an integer linear programming model for the one-dimensional cutting stock and scheduling problem with heterogeneous orders. As this problem is a generalization of the classical single-period one-dimensional CSP, which is known to be NP-hard, it is difficult to solve real-sized instances to optimality. Thus, we propose a novel matheuristic algorithm based on a fix-and-optimize strategy hybridized with a random local search. The proposed matheuristic was tested on a set of 160 synthetic problem instances based on a real-world problem and compared with CPLEX solver. In larger instances, the proposed matheuristic performed better than CPLEX, with average relative percentage deviation (RPD) regarding objective values as high as 72%. On the other hand, in small instances CPLEX showed a marginal advantage, with best average RPD of 18% with relation to the matheuristic. We also performed a paired t-test with significance level 0.05 and null hypothesis: no difference between the proposed matheuristic and CPLEX. In small test instances, the performance of the proposed matheuristic was statistically indistinguishable from CPLEX, while in larger instances the matheuristic outperformed CPLEX in most cases with p value < 0.05.

Keywords Cutting and packing problems · Order scheduling problems · Fix-and-optimize · Matheuristics

✉ Anselmo Ramalho Pitombeira-Neto
anselmo.pitombeira@ufc.br

Bruno de Athayde Prata
baprata@ufc.br

1 Department of Industrial Engineering, Federal University of Ceará, Campus do Pici, Fortaleza, Brazil

Mathematics Subject Classification 90C10 · 90B35 · 90C59

1 Introduction

Cutting stock and bin packing problems are common in several industrial sectors, such as paper, glass, furniture, steel industry, construction, transportation, among others. They belong to the class of combinatorial optimization problems and have received a lot of attention from researchers in the last few decades, mainly due to them being NP-hard. Therefore, obtaining optimal solutions in acceptable computational times is a difficult task when solving large real-world problem instances. Dyckhoff (1990) and Wäscher et al. (2007) present comprehensive classifications of cutting and packing problems.

The classical cutting stock problem (CSP) ignores the production planning and scheduling of multiple customer orders. Nevertheless, in real industrial settings customer orders have to be planned and scheduled over time so as to meet demand and required due dates. This gives rise to combined cutting stock and production planning problems, whose objective is to solve the CSP over multiple time periods. While the literature on classical CSP is abundant, interest in combined cutting stock and production planning problems is more recent.

The integration of the CSP and production planning has been taking two directions in the literature. In the first one, the CSP is extended to a multiperiod setting in which at a given time period a set of customer orders is fulfilled and inventory is hold to next time periods. The objective is the minimization of a weighed sum of production, trim loss and inventory costs. This is often referred to as the combined cutting stock and lot sizing problem, which is more common in make-to-stock production environments and has been studied in Nonås and Thorstenson (2000), Gramani and França (2006), Nonås and Thorstenson (2008), Trkman and Gradisar (2007), Poldi and de Araujo (2016), Melega et al. (2018). In contrast, in the second direction the CSP is extended to a multiperiod setting in which customer orders are fulfilled in a make-to-order fashion. An order may take several time periods to be completed, so that scheduling of multiple orders is relevant. The objective in this case is the minimization of trim loss from the cutting process and minimization of a criterion related to order scheduling such as the makespan or total tardiness. In this paper, we are interested in this second class of problems, known as *cutting stock and scheduling problems* (CSSP).

One of the first approaches addressing the integration of cutting stock and scheduling problems was presented by Yuen (1991), who proposed a heuristic for sequencing cutting patterns. Li (1996) presented a model for a multistage two-dimensional CSP considering due dates and release dates, in which the orders are scheduled before starting the cutting process. Arbib and Marinelli (2005) proposed an integer linear programming model that integrates the cutting process in a first production stage and the assembly of parts in a second stage. Yanasse (1997); Yanasse and Lamosa (2007) developed an integer programming model for pattern sequencing considering the minimization of the number of open stacks. Reinertsen and Vossen (2010) proposed a model for scheduling patterns in the one-dimensional CSP with due dates, whose solution method was further improved by Arbib and Marinelli (2014); Braga et al. (2016a, b) proposed models for the scheduling of customer orders in the one-dimensional CSP with due dates.

Most models proposed so far approach the CSSP with homogeneous orders, i.e., they assume that all demanded items in an order are of the same type. For example, if an item type is defined by a width, all items in an order have the same width. However, motivated by the problem of precast beam production considered in Prata et al. (2015), we noticed that in many industrial settings this assumption is not realistic, since a customer order may include items of multiple types. We then approach in this paper the one-dimensional CSSP with heterogeneous orders. In this problem, a set of heterogeneous orders demanded by customers has to be met in a make-to-order fashion. Each order takes many time periods to be completed and orders may be processed concurrently. The objective is to minimize total tardiness while minimizing trim loss from the cutting process.

The main contributions of this paper are twofold. First, we introduce an extension to Braga et al. (2016a)'s one-dimensional CSSP compact model, which assumes homogeneous orders, for the case with heterogeneous orders. Second, as this problem is a generalization of the classical single-period one-dimensional CSP, which is known to be NP-hard (Martello 1990), it is difficult to solve real-sized instances to optimality. Thus, we propose a novel matheuristic algorithm based on a fix-and-optimize strategy hybridized with a random local search. Matheuristic algorithms blend mathematical programming-based methods with (meta)heuristics in order to obtain good solutions to hard optimization problems. It has recently attracted attention from many researchers in the field, see for example Ozer and Sarac (2018), Archetti and Speranza (2014), Kramer et al. (2015), Della Croce and Salassa (2014), Melo et al. (2014), Lin and Ying (2016), Miranda et al. (2018). The proposed matheuristic algorithm is tested on a set of synthetic instances and compared with CPLEX solver.

The remainder of this paper is organized as follows: in Sect. 2, we formulate the proposed integer linear programming; in Sect. 3, we describe the proposed matheuristic algorithm; in Sect. 4, we describe the computational experiment design and discuss the results; finally, in Sect. 5, we present some conclusions and suggestions for future work.

2 Problem formulation

In this section, we propose a compact model based on the one described in Braga et al. (2016a, 2015). Our model is a generalization of the one we proposed for a scheduling problem in the precast beams industry, which is described in the paper Prata et al. (2015).

Let $\mathscr{K} = \{1, 2, \ldots, K\}$ be a set of customer orders with $K = |\mathscr{K}|$ and $\mathscr{I} = \{1, 2, \ldots, I\}$ a set of item types with lengths $l_1, l_2, \ldots, l_I$ and $I = |\mathscr{I}|$. For each order $k \in \mathscr{K}$, there is a vector $\mathbf{n}_k = (n_{1k}, n_{2k}, \ldots, n_{Ik}) \in \mathbb{Z}^I_{\geq 0}$, where n_{ik} is the demand for item type i in order k. In addition, each order k has a due date d_k, which is a future time period past which the order is late. These orders have all to be completed within a discrete planning horizon of T time periods. At each time period $t = 1, 2, \ldots, T$, there are J objects of size L available in stock from which the demanded items may be cut. In contrast to Braga et al.'s model, we do not assume that only one object is cut in a time period, but that J objects are available to be cut. In addition, while in Braga et al.'s model, it is assumed that each order is composed of items of the same type; in our model, we relax this assumption and allow an order to include items of different types, hence the term *heterogeneous orders*.

We then define the following decision variables: $x_{ikjt} \in \mathbb{Z}_{\geq 0}$ is the number of items of type i from order k cut from object j at time t, where $\mathbb{Z}_{\geq 0}$ denotes the set of non-negative integers; $w_{jt} \in \{0, 1\}$ in which 1 indicates whether object j is used at time t, 0 otherwise; and $z_{kt} \in \{0, 1\}$ in which 1 indicates whether order k has items produced at time t and 0 otherwise, for $t = d_k + 1, \ldots, T$. The latter variables are used to compute the tardiness of the corresponding orders. For convenience, we organize below the notation used:

Indices and sets

i: index for item types in $\mathscr{I} = \{1, 2, \ldots, I\}$;
k: index for customer orders in $\mathscr{K} = \{1, 2, \ldots, K\}$;
j: index for objects, $j \in \{1, 2, \ldots, J\}$;
t: index for time periods, $t \in \{1, 2, \ldots, T\}$;

Parameters

n_{ik}: demand for item type i in order k;
l_i: length of item i;
d_k: due date of the order k;
L: size of available objects in stock;

Decision variables

$x_{ikjt} \in \mathbb{Z}_{\geq 0}$: number of items of type i from order k cut from object j at time t;

$$w_{jt} = \begin{cases} 1, & \text{if object } j \text{ is used at time } t; \\ 0, & \text{otherwise;} \end{cases}$$

$$z_{kt} = \begin{cases} 1, & \text{if the order } k \text{ has items produced at time } t; \\ 0, & \text{otherwise.} \end{cases}$$

The proposed integer linear programming model (hereafter called model M) is formulated as follows:

$$\min \sum_{k=1}^{K} \sum_{t=d_k+1}^{T} z_{kt} + \sum_{j=1}^{J} \sum_{t=1}^{T} w_{jt} \tag{1}$$

$$\text{s.t.} \sum_{j=1}^{J} \sum_{t=1}^{T} x_{ikjt} = n_{ik} \quad i = 1, \ldots, I, \ k = 1, \ldots, K, \tag{2}$$

$$\sum_{i=1}^{I} \sum_{k=1}^{K} l_i x_{ikjt} \leq L w_{jt} \quad j = 1, \ldots, J, \ t = 1, \ldots, T, \tag{3}$$

$$\sum_{j=1}^{J} x_{ikjt} + z_{k(t+1)} \leq M_{ik} z_{kt} \quad i=1,\ldots,I,\; k=1,\ldots,K,\; t=d_k+1,\ldots,T, \tag{4}$$

$$x_{ikjt} \in \mathbb{Z}_{\geq 0} \quad i=1,\ldots,I,\; j=1,\ldots,J,\; t=1,\ldots,T, \tag{5}$$

$$z_{kt} \in \{0,1\} \quad k=1,\ldots,K,\; t=d_k+1,\ldots,T, \tag{6}$$

$$w_{jt} \in \{0,1\} \quad j=1,\ldots,J,\; t=1,\ldots,T. \tag{7}$$

The objective function (1) includes a first term that represents the total tardiness of the orders and a second term that corresponds to the number of objects in stock needed to meet the demand; constraints (2) require that all items demanded by all orders are produced; constraints (3) enforce that the total size of the items cut from an object is not greater than its size; constraints (4) assure that if an item demanded by an order is produced at time $t \geq d_k + 1$, then that time period is added to its tardiness. [We define $z_{k(T+1)} = 0$ for $k = 1, \ldots, K$ to maintain consistency of constraints (4)]. Finally, constraints (5), (6) and (7) define the domain of the decision variables. The constants M_{ik} may be set as upper bounds on the left-hand side of constraints (4) according to

$$M_{ik} = \min(n_{ik} + 1, J\lfloor L/l_i \rfloor + 1) \quad i=1,\ldots,I,\; k=1,\ldots,K. \tag{8}$$

In our model, we assume that all J objects available at a time period are used and no objects are kept in inventory for future use. Proposition 1 shows that, if the model has an optimal solution, it is possible to obtain an optimal solution in which no inventory is kept. In this way, we can constrain our feasible set to include only solutions with no inventory. This property stems from the fact the our objective function involves minimizing the total tardiness, and keeping inventory contributes to delaying the completion of orders.

Proposition 1 *If model* M *has an optimal solution, it is possible to obtain an optimal solution in which no inventory is kept.*

Proof Let s_1 be an optimal solution and v_1 its objective value given by equation (1) in the model. In addition, let $t^{\max} = \max_{k=1,\ldots,K}\{\sum_{t=1}^{T} z_{kt}\}$ be the makespan of all orders in s_1, so that $z_{kt} = 0$ for $t = t^{\max} + 1$, $k = 1, \ldots, K$. Further assume that at a given time $t' < t^{\max}$ there is some unused object p for which $w_{pt'} = 0$, i.e., this object was kept in inventory. As the completion of all orders finish at $t^{\max}$, there is at least an object q for which $w_{qt^{\max}} = 1$. Let s_2 be a solution obtained from s_1 by making $w_{pt'} = 1$, $w_{qt^{\max}} = 0$ and keeping unchanged all other variables. In s_2, the object p is used and not kept in inventory. We use superscripts (1) and (2) to denote values in solutions s_1 and s_2, respectively. In the new solution s_2, constraint (3) for object q at time $t^{\max}$ is modified so that

$$\sum_{i=1}^{I} \sum_{k=1}^{K} l_i x_{ikqt^{\max}}^{(2)} \leq 0, \tag{9}$$

which implies $x^{(2)}_{ikqt^{\max}} = 0$ for $i = 1, \ldots, I$, $k = 1, \ldots, K$. In contrast, for object p at time t' we have

$$\sum_{i=1}^{I}\sum_{k=1}^{K} l_i x^{(2)}_{ikpt'} \leq L. \tag{10}$$

Feasibility in constraint (2) can be maintained by making $x^{(2)}_{ikpt'} = x^{(1)}_{ikqt^{\max}}$ for $i = 1, \ldots, I$, $k = 1, \ldots, K$. In addition, we notice that feasibility in constraint (4) at time $t^{\max}$ is maintained, since from constraint (9)

$$\sum_{j=1}^{J} x^{(2)}_{ikqt^{\max}} \leq \sum_{j=1}^{J} x^{(1)}_{ikqt^{\max}}, \tag{11}$$

so that

$$\sum_{j=1}^{J} x^{(2)}_{ikqt^{\max}} + z^{(2)}_{k(t^{\max}+1)} \leq M_{ik} z^{(2)}_{kt^{\max}}. \tag{12}$$

Also at time t' feasibility in constraint (4) is maintained, since

$$\sum_{j=1}^{J} x^{(2)}_{ikpt'} + z^{(2)}_{k(t'+1)} \leq M_{ik} z^{(2)}_{kt'}, \tag{13}$$

as $t' < t^{\max}$ and $z^{(2)}_{kt'} = 1$ for at least some order k. We then notice that objective value $v_2 = v_1$, since the only change was $w_{pt'} = 1$, $w_{qt^{\max}} = 0$, so that $\sum_{k=1}^{K}\sum_{t=d_k+1}^{T} z^{(2)}_{kt} = \sum_{k=1}^{K}\sum_{t=d_k+1}^{T} z^{(1)}_{kt}$ and $\sum_{j=1}^{J}\sum_{t=1}^{T} w^{(2)}_{jt} = \sum_{j=1}^{J}\sum_{t=1}^{T} w^{(1)}_{jt}$.

Finally, we have an optimal solution s_2 so that object p at time t' which was kept in inventory is now used. In case this was the only object kept in inventory, now s_2 is an optimal solution with no inventory. Otherwise, we can repeat the above process and obtain successive optimal solutions $s_3, s_4 \ldots s_N$, in which N is the number of objects held in inventory until no object is kept in inventory anymore. $\square$

The number of variables and constraints in the model, as a function of the set sizes is shown in Table 1. In some preliminary experiments, we have observed that the performance of solvers greatly decreases as the number of variables increases. In particular, scalars J and T have the largest impact on the size of the model. Set size J is a feature of the problem instance which one intends to solve. On the other hand, the

Table 1 Number of variables and constraints in the model as a function of set sizes

Parameter	Value
#Variables	$IKJT + JT + \sum_{k=1}^{K}(T - d_k - 1)$
#Constraints	$IK + JT + \sum_{i=1}^{I}\sum_{k=1}^{K}(T - d_k - 1)$

planning horizon T must be specified by the modeler. The greater the horizon, the more variables there are in the model. Given a set of orders, we should specify T as small as possible in order to not have too much variables. On the other hand, if T is too small there may be no feasible solution. Proposition 2 below gives an upper bound on T.

Proposition 2 *An upper bound on T is given by*

$$T^{\text{UB}} = \left\lceil \frac{\sum_{i=1}^{I} \sum_{k=1}^{K} l_i n_{ik}}{J(L - \max\{l_1, l_2, \ldots, l_I\})} \right\rceil. \tag{14}$$

Proof Without loss of generality, assume for a given feasible solution that the total time to finish all orders is T and that all objects J at time $t = 1, 2, \ldots, T$ are used. In addition, assume that the maximum trim loss at any used object is $\max\{l_1, l_2, \ldots, l_I\}$, otherwise we could just use the trim loss to cut an item with this maximum length. The total length of objects used to fulfill the demand is

$$\sum_{i=1}^{I} \sum_{k=1}^{K} l_i n_{ik} = \sum_{t=1}^{T} \sum_{j=1}^{J} (L - e_{jt}), \tag{15}$$

in which e_{jt} is the trim loss of object j at time t. Dividing by TJ, we have

$$\frac{1}{TJ} \sum_{i=1}^{I} \sum_{k=1}^{K} l_i n_{ik} = L - \frac{1}{TJ} \sum_{t=1}^{T} \sum_{j=1}^{J} e_{jt}. \tag{16}$$

Let

$$\bar{e} = \frac{1}{TJ} \sum_{t=1}^{T} \sum_{j=1}^{J} e_{jt}, \tag{17}$$

be the average trim loss. Then, from (16) and (17) we have

$$T \leq \left\lceil \frac{\sum_{i=1}^{I} \sum_{k=1}^{K} l_i n_{ik}}{J(L - \bar{e})} \right\rceil, \tag{18}$$

and as $\bar{e} \leq \max\{l_1, l_2, \ldots, l_I\}$, then

$$T \leq \left\lceil \frac{\sum_{i=1}^{I} \sum_{k=1}^{K} l_i n_{ik}}{J(L - \max\{l_1, l_2, \ldots, l_I\})} \right\rceil . \square \tag{19}$$

2.1 An illustrative example

In this section, we solve a small problem instance to illustrate the application of our proposed model and to validate it. Consider that a manufacturer offers seven different types of items with lengths $\mathscr{I} = \{1.22, 1.45, 2.35, 2.50, 2.65, 2.95, 3.30\}$. At each time period, there are $J = 5$ objects available in stock, with $L = 11.95$.

At a certain moment, the manufacturer receives $k = 3$ orders from three different customers, with the following demands: order 1 = (4, 10, 9, 17, 1, 7, 2), order 2 = (4, 10, 12, 14, 5, 5, 0), order 3 = (4, 16, 7, 15, 3, 4, 1), and due dates 6, 6 and 7, respectively. From Eq. (14), a upper bound on the total time to finish all three orders is $T^{\mathrm{UB}} = 19$. In an Intel Core i5 machine with 2.3GHz and 8 GB RAM, CPLEX 12.8 solves this small instance to optimality in less than 20 seconds. In the optimal solution, 28 objects are used to satisfy the three orders in 7 time periods with no tardy order.

3 Proposed solution algorithm

We propose a matheuristic algorithm in which model M is iteratively solved in search for better integer solutions in which a subset of variables have fixed values. We notice that z_{kt} are complicating variables in M. If we fix these variables, the resulting model is easier than the full model. Our idea is to predetermine the completion times of the orders and fix the corresponding z_{kt} variables. We then solve the resulting model in which we try to fit all demanded items within the available objects limited by the predetermined completion times. We repeat this process with different completion times determined by a heuristic procedure.

Let $\mathscr{C} = \{c_1, c_2, \ldots, c_K\}$ be a set of completion times of the orders in $\mathscr{K}$. We denote by $\mathrm{M}_{\mathscr{C}}$ the model M in which the variables z_{kt} are fixed at values according to the completion times in $\mathscr{C}$. We note that, following constraint (4) in M, for $t = d_k, \ldots, T$ and $k = 1, \ldots K$, $z_{kt} = 1$ if $t \leq c_k$ and $z_{kt} = 0$ otherwise. For given $\mathscr{C}$, we solve $\mathrm{M}_{\mathscr{C}}$ with a probing time limit τ, with the purpose of finding a feasible solution. In general, the solver finds a feasible solution very fast or identifies that $\mathrm{M}_{\mathscr{C}}$ is infeasible, but in some cases it may spend too much time to reach one of these two states. The probing time τ is then a parameter set by the analyst corresponding to the maximum time she is willing to assign to the solver in order check feasibility.

If $\mathrm{M}_{\mathscr{C}}$ turns out to be feasible, we denote respectively by $s(\mathrm{M}_{\mathscr{C}}; \tau)$ and $v(\mathrm{M}_{\mathscr{C}}; \tau)$ the best integer solution and its corresponding objective value under probing time τ. We then sample an order o from the discrete uniform distribution with support in $\{1, 2, \ldots, K\}$, i.e. $o \sim \mathrm{DiscUnif}(1, K)$, and decrease its completion time by $\delta \sim \mathrm{DiscUnif}(\delta_{\min}, \delta_{\max})$. Otherwise, if $\mathrm{M}_{\mathscr{C}}$ turns out to be infeasible or feasibility could not be checked before time τ, we sample $o \sim \mathrm{DiscUnif}(1, K)$ and increase its completion time by $\delta \sim \mathrm{DiscUnif}(\delta_{\min}, \delta_{\max})$. We repeat this process for a predefined number of iterations N, with τ, $\delta_{\min}$ and $\delta_{\max}$ as specified parameters. Algorithm 1 details our proposed matheuristic.

4 Computational results

4.1 Problem instances

Synthetic problem instances were generated based on real data from Prata et al. (2015), in which the authors tackle the problem of multiperiod production planning of precast beams. In the following paragraphs, we will refer to these as PPS

Algorithm 1

```
procedure MATHEURISTIC
    Set starting 𝒞^(0) ← {c_1^(0), c_2^(0), ..., c_K^(0)}        ▷ Initial set of completion times
    v_best ← +∞
    s_best ← ∅
    for i ← 1 ... N do
        for k ← 1 ... K do                                      ▷ Fix variables z_kt
            for t ← d_k ... T do
                if t ≤ c_k^(i−1) then
                    z_kt ← 1
                else
                    z_kt ← 0
                end if
            end for
        end for
        Solve model M_{𝒞^(i−1)} with probing time limit τ
        if M_{𝒞^(i−1)} is feasible then
            if v(M_{𝒞^(i−1)}; τ) < v_best then
                v_best ← v(M_{𝒞^(i−1)}; τ)
                s_best ← s(M_{𝒞^(i−1)}; τ)
            end if
            Sample o ∼ DiscUnif(1, K)
            Sample δ^(i) ∼ DiscUnif(δ_min, δ_max)
            c_o^(i) ← c_o^(i−1) − δ^(i)                          ▷ Decrease completion time of order o
        else                                                    ▷ M is infeasible or undefined
            Sample o ∼ DiscUnif(1, K)
            Sample δ^(i) ∼ DiscUnif(δ_min, δ_max)
            c_o^(i) ← c_o^(i−1) + δ^(i)                          ▷ Increase completion time of order o
        end if
        for k ← 1 ... K do
            if k ≠ o then
                c_k^(i) ← c_k^(i−1)
            end if
        end for
        𝒞^(i) ← {c_1^(i), c_2^(i), ..., c_K^(i)}                 ▷ Update set of completion times
    end for
end procedure
```

data (from Prata, Pitombeira-Neto and Sales, the authors from which the data were taken.) In the PPS data, there is only one order with $I = 7$ item types, corresponding lengths given by the set $\mathscr{I} = \{1.22, 1.45, 2.35, 2.50, 2.65, 2.95, 3.30\}$ and demands $n^{\text{PPS}} = (24, 60, 56, 72, 16, 17, 12)$, respectively, with a total of $N^{\text{PPS}} = 257$ items demanded. The items must be packed into objects of size $L = 11.95$.

In the real problem from which these data originated, the demands for each item type vary a lot among orders, while the item types and object sizes are kept constant by engineering design. We then generated 8 instance sets in which we varied two factors: the number of orders $K \in \{5, 10, 20, 30\}$ and the number of objects $J \in \{5, 10\}$. These levels correspond to values which are likely to occur in reality. For each level of K and J, we generated 20 instances in which we sampled the demands in the following manner: for each order $k \in \{1, 2, \ldots, K\}$, we first sampled the total order demand $N_k \sim \text{DiscUnif}(100, 500)$ from a discrete uniform distribution. We then shared the total demand among the item types $i \in \mathscr{I}$ according to a multinomial distribution, i.e., $n_{ik} \sim \text{Multi}(N_k, \mathbf{p})$ in which $\mathbf{p} = (p_1, p_2, \ldots, p_I)$ and p_i is the probability that an item of type i is demanded by a customer order. We estimated p_i by $\hat{p}_i = n_i^{\text{PPS}}/N^{\text{PPS}}$. The due dates for each order $k \in 1, 2, \ldots, K$ were sampled according to $d_k \sim \text{DiscUnif}(\lceil T/4 \rceil, T)$ with T given by (14).

4.2 Experimental parameters

For each of the 160 generated instances, we applied both IBM ILOG CPLEX solver version 12.8 and the proposed matheuristic given by Algorithm 1. We used an Intel Core i7-7700K machine with 4.2 GHz clock and 8GB RAM. We set a time limit of 3600s for CPLEX.

In early experiments, we noticed that the CPLEX branch-and-cut (B&C) tree filled up the whole RAM before reaching the time limit. To overcome this hurdle, we set the CPLEX nodefile parameter to 3, so that the B&C tree could be stored in the solid-state drive when it reached a critical size. In some instances, CPLEX filled up the RAM still before starting the B&C tree. We then had to disable multithreading in these instances and ran only one thread.

Regarding our proposed matheuristic, we adopted parameters $\delta_{\min} = 1$, $\delta_{\max} = 10$, $\tau = 60s$ and $N = 50$ iterations. Notice that this gives a total time of $N \times \tau = 3000$s. After that, we started warm CPLEX with the best found solution by the matheuristic and ran it with a time limit of 600s to possibly obtain a better solution and compute its gap, which gives a total running time of $3000+600 = 3600$s. Initial completion times of all orders are arbitrarily set to be equal to the midpoint of the planning horizon T of the particular problem instance, i.e $c_1^{(0)} = c_2^{(0)} =, \ldots, = c_K^{(0)} = T/2$. Notice that this implies no use of prior information on feasible values for the completion times of orders. Alternatively, one could first apply a constructive heuristic to find initial feasible completion times.

4.3 Performance measures

We used as performance measures the relative percentage deviation (RPD), the linear relaxation gap (LRG) and the percentage of instances (WINS) in which the matheuristic obtained a solution better than the one obtained by CPLEX. RPD is defined as

$$\text{RPD} = \frac{v_{\text{M}} - v_{\text{C}}}{v_{\text{C}}} \times 100,$$

where v_{M} denotes the best objective function value obtained by the matheuristic, while v_{C} denotes the best objective function value obtained by CPLEX. LRG is defined as

$$\text{LRG} = \frac{UB - LB}{UB} \times 100,$$

where UB is the upper bound from CPLEX after running for the specified time and not finding the optimal solution, corresponding to the objective value of the incumbent solution, and LB is the lower bound corresponding to the objective value of a linear relaxation of a subproblem in the B&C tree. WINS is defined as

$$\text{WINS} = \frac{n_{\text{M}}}{n} \times 100,$$

where n_{M} is the number of instances in which the matheuristic achieved a better solution than CPLEX and n is the number of instances in the given instance set.

In addition, to evaluate if there is in fact a statistically significant difference between the proposed matheuristic and CPLEX, we applied a paired t test on each instance set with a null hypothesis that there is no difference between mean values from the solutions obtained by the proposed matheuristic and CPLEX in each instance set. The paired t test compares two samples resulting from two experiments applied to the same units. Montgomery (2013) Typically, the two samples refer to responses of the applications of two different so-called "treatments" in the nomenclature of statistical testing. In our case, the units are the problem instances, the responses are the objective values of the instances, and the two treatments are the tested algorithms, namely: our proposed matheuristic and the branch-and-cut algorithm implemented in CPLEX. The null hypothesis is that there is no difference between the means of the two treatments, while the alternative hypothesis is that there is such a difference. The application of the t-test gives us statistical evidence if there is in fact a difference between the two algorithms or if the observed difference is due to randomness in the experiments.

4.4 Results and discussion

Tables 2 and 3 show the results for both CPLEX and the proposed matheuristic for each instance set, while Figs. 1 and 2 show results for each of the 160 instances. (Test instances and results may be obtained from the authors upon request.) It is worth noticing that neither CPLEX nor the matheuristic was able to find an optimal solution to any instance in less than the specified time of 3600s, which is an empirical evidence of the computational difficulty of the problem. In fact, in preliminary experiments, the largest instance we could solved to optimality within a time limit of 3600 s, and which involved a nontrivial number of demanded items, had only 1 bin and 2 orders.

From Table 2, it can be seen that the proposed matheuristic obtained on average better solutions than CPLEX. In instance sets 3, 4, 7 and 8, the matheuristic generated

Table 2 Comparison results between CPLEX and the proposed matheuristic (20 instances per set)

Instance set	#Bins	#Orders	Avg. RPD (%)	Std. Dev. RPD (%)	WINS (%)
1	5	5	0.39	1.02	30
2	5	10	18.04	39.72	15
3	5	20	− 54.68	55.54	95
4	5	30	− 71.97	2.55	100
5	10	5	0.02	0.44	40
6	10	10	4.34	17.45	15
7	10	20	− 63.20	16.68	100
8	10	30	− 67.43	5.02	100

Table 3 Solution gaps obtained from CPLEX and the proposed matheuristic (20 instances per set)

Instance set	#Bins	#Orders	Avg. Gap Math (%)	Avg. Gap CPLEX (%)
1	5	5	2.23	1.86
2	5	10	24.23	13.75
3	5	20	65.77	84.35
4	5	30	76.03	93.28
5	10	5	1.84	1.83
6	10	10	11.18	7.31
7	10	20	46.55	80.36
8	10	30	60.91	87.27

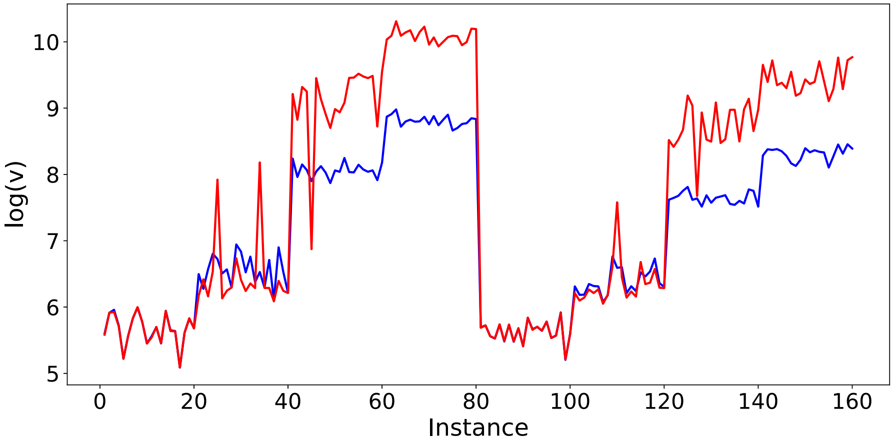

Fig. 1 Logarithm of objective values of best solutions obtained by the proposed matheuristic (blue line) and CPLEX (red line) for all 160 instances (color figure online)

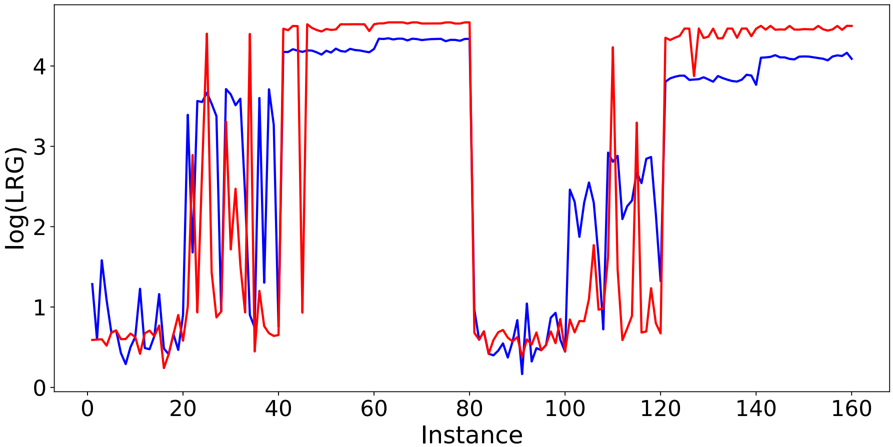

Fig. 2 Logarithm of LRG of best solutions obtained by the proposed matheuristic (blue line) and CPLEX (red line) for all 160 instances (color figure online)

Table 4 Results from paired t test (H_0: no difference between the proposed matheuristic and CPLEX)

Instance set	#Bins	#Orders	t-statistic	p value	Reject? ($p < 0.05$)
1	5	5	1.5304	1.4224×10^{-1}	No
2	5	10	– 0.5421	5.9378×10^{-1}	No
3	5	20	– 8.3503	5.9748×10^{-8}	Yes
4	5	30	– 29.472	5.9385×10^{-18}	Yes
5	10	5	– 0.9999	3.2930×10^{-1}	No
6	10	10	– 0.3152	7.5586×10^{-1}	No
7	10	20	– 8.6602	3.3518×10^{-8}	Yes
8	10	30	– 15.84	8.7285×10^{-13}	Yes

significantly better solutions than CPLEX. These are sets with 20 and 30 orders, which include the hardest instances. In instance sets 1, 5 and 6, its performance was comparable with CPLEX. The noteworthy exception was instance set 2, in which the performance of the matheuristic was significantly worse. In addition, it can be seen that instance sets 2, 3, 6 and 7 have considerably higher standard deviations. We identified that instance set 2 has really dispersed results, while instance sets 3, 6 and 7 each present outlier instances which inflated the standard deviation. Removing the one outlier instance in instance sets 3, 6 and 7, we obtained the values 8.08%, 7.45% and 9.20% for the standard deviations, respectively.

To evaluate if there is in fact a statistically significant difference between the proposed matheuristic and CPLEX, we applied a paired t-test on each instance set with a null hypothesis that there is no difference between mean values from the solutions obtained by the proposed matheuristic and CPLEX. Assuming a significance level of 0.05, the corresponding p values indicate that there is a statistically significant difference in instance sets 3, 4, 7 and 8. (See Table 4) In all these 4 instance sets, the proposed matheuristic achieved better average result than CPLEX, according to both average RPD and WINS (See Table 2). In contrast, there is no statistically significant difference in instance sets 1, 2, 5 and 6, so that we do not have evidence to state there is any performance difference between the proposed matheuristic and CPLEX in these instance sets.

Finally, as a comparison basis, we also obtained a lower bound for each instance by solving model M (Eqs. (1)–(7)) in CPLEX with only variables x_{ijkt} relaxed, i.e., these variables were allowed to assume fractional values. The resulting model is now a mixed integer programming model, which could be solved to optimality in most instances within a time limit of 3600 s. We call this lower bound LB*. Table 5 shows the relative deviation of the objective value f_{obj} obtained from CPLEX and from the proposed matheuristic to LB* computed as $(f_{\text{obj}} - \text{LB}^*)/\text{LB}^*$.

5 Final remarks

In this paper, we addressed the one-dimensional cutting stock and scheduling problem with heterogeneous orders in which we have customer orders composed of heteroge-

Table 5 Relative deviations to LB* (averages over instances solved to optimality)

Instance set	#Bins	#Orders	Matheuristics	CPLEX	#Instances solved
1	5	5	0.0211	0.0172	20
2	5	10	0.3689	0.5042	20
3	5	20	1.9238	8.0257	20
4	5	30	3.0202	13.496	19
5	10	5	0.0174	0.0172	20
6	10	10	0.1277	0.1540	20
7	10	20	0.8750	4.7851	13
8	10	30	1.5356	7.0068	9

neous types of items. The objective function to be minimized is a linear combination of total tardiness and the number of objects in stock needed to meet the demand. For this NP-hard problem, we proposed an integer linear programming model and a novel matheuristic algorithm based on a fix-and-optimize strategy hybridized with a random local search.

Extensive computational experiments were carried out in which we compared the performance of the proposed matheuristic with IBM CPLEX. In small-scale test instances, the performance of the proposed matheuristic was statistically indistinguishable from CPLEX, while in larger instances the matheuristic outperformed CPLEX in most cases with statistically significant results. ($p < 0.05$)

As future studies, we recommend the use of size-reduction heuristics, as proposed by Fanjul-Peyro and Ruiz (2011), to improve the solutions generated by the proposed matheuristic. Furthermore, other objective functions may be considered, such as minimization of the total completion time or the makespan.

Acknowledgements The authors would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq; Grant No.: 422464/2016-3) for the financial support.

References

Arbib C, Marinelli F (2005) Integrating process optimization and inventory planning in cutting-stock with skiving option: an optimization model and its application. Eur J Oper Res 163(3):617–630. https://doi.org/10.1016/j.ejor.2003.12.021

Arbib C, Marinelli F (2014) On cutting stock with due dates. Omega 46:11–20. https://doi.org/10.1016/j.omega.2014.01.004

Archetti C, Speranza MG (2014) A survey on matheuristics for routing problems. EURO J Comput Optim 2(4):223–246. https://doi.org/10.1007/s13675-014-0030-7

Braga N, Alves C, Macedo R, de Carvalho JV (2015) A model-based heuristic for the combined cutting stock and scheduling problem. In: Gervasi O et al (eds) Computational science and its applications – ICCSA 2015. ICCSA 2015. Lecture Notes in Computer Science, vol 9156. Springer, Cham

Braga N, Alves C, de Carvalho J (2016a) Exact solution of combined cutting stock and scheduling problems. Lect Notes Econ Math 682:131–139

Braga N, Alves C, Macedo R, de Carvalho JV (2016b) Combined cutting stock and scheduling: a matheuristic approach. Int J Innov Comput Appl 7(3):135–146. https://doi.org/10.1504/IJICA.2016.078724

Della Croce F, Salassa F (2014) A variable neighborhood search based matheuristic for nurse rostering problems. Ann Oper Res 218(1):185–199. https://doi.org/10.1007/s10479-012-1235-x
Dyckhoff H (1990) Cutting and packing a typology of cutting and packing problems. Eur J Oper Res 44(2):145–159. https://doi.org/10.1016/0377-2217(90)90350-K
Fanjul-Peyro L, Ruiz R (2011) Size-reduction heuristics for the unrelated parallel machines scheduling problem. Comput Oper Res 38(1):301–309. https://doi.org/10.1016/j.cor.2010.05.005
Gramani MCN, França PM (2006) The combined cutting stock and lot-sizing problem in industrial processes. Eur J Oper Res 174(1):509–521. https://doi.org/10.1016/j.ejor.2004.12.019
Kramer R, Subramanian A, Vidal T, Cabral LdAF (2015) A matheuristic approach for the pollution-routing problem. Eur J Oper Res 243(2):523–539. https://doi.org/10.1016/j.ejor.2014.12.009
Li S (1996) Multi-job cutting stock problem with due dates and release dates. J Oper Res Soc 47(4):490–510. https://doi.org/10.1057/jors.1996.56
Lin SW, Ying KC (2016) Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. Omega 64:115–125. https://doi.org/10.1016/j.omega.2015.12.002
Martello S (1990) Knapsack problems: algorithms and computer implementations. Wiley-Interscience series in discrete mathematics and optimiza tion
Melega GM, de Araujo SA, Jans R (2018) Classification and literature review of integrated lot-sizing and cutting stock problems. Eur J Oper Res 271(1):1–19. https://doi.org/10.1016/j.ejor.2018.01.002
Melo MT, Nickel S, Saldanha-da Gama F (2014) An efficient heuristic approach for a multi-period logistics network redesign problem. TOP 22(1):80–108. https://doi.org/10.1007/s11750-011-0235-3
Miranda PL, Morabito R, Ferreira D (2018) Optimization model for a production, inventory, distribution and routing problem in small furniture companies. TOP 26(1):30–67. https://doi.org/10.1007/s11750-017-0448-1
Montgomery DC (2013) Design and analysis of experiments, 8th edn. Wiley, Hoboken
Nonås SL, Thorstenson A (2000) A combined cutting-stock and lot-sizing problem. Eur J Oper Res 120(2):327–342. https://doi.org/10.1016/S0377-2217(99)00160-5
Nonås SL, Thorstenson A (2008) Solving a combined cutting-stock and lot-sizing problem with a column generating procedure. Comput Oper Res 35(10):3371–3392. https://doi.org/10.1016/j.cor.2007.03.005
Ozer EA, Sarac T (2018) MIP models and a matheuristic algorithm for an identical parallel machine scheduling problem under multiple copies of shared resources constraints. TOP pp 1–31, https://doi.org/10.1007/s11750-018-00494-x
Poldi K, de Araujo S (2016) Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem. Ann Oper Res 238(1–2):497–520. https://doi.org/10.1007/s10479-015-2103-2
Prata BA, Pitombeira-Neto AR, Sales CJM (2015) An integer linear programming model for the multiperiod production planning of precast concrete beams. J Constr Eng Manag 141(10):04015029. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000991
Reinertsen H, Vossen T (2010) The one-dimensional cutting stock problem with due dates. Eur J Oper Res 201(3):701–711. https://doi.org/10.1016/j.ejor.2009.03.042
Trkman P, Gradisar M (2007) One-dimensional cutting stock optimization in consecutive time periods. Eur J Oper Res 179(2):291–301. https://doi.org/10.1016/j.ejor.2006.03.027
Wäscher G, HauSSner H, Schumann H (2007) An improved typology of cutting and packing problems. Eur J Oper Res 183(3):1109–1130. https://doi.org/10.1016/j.ejor.2005.12.047
Yanasse H (1997) On a pattern sequencing problem to minimize the maximum number of open stacks. Eur J Oper Res 100(3):454–463. https://doi.org/10.1016/S0377-2217(97)84107-0
Yanasse HH, Lamosa MJP (2007) An integrated cutting stock and sequencing problem. Eur J Oper Res 183(3):1353–1370. https://doi.org/10.1016/j.ejor.2005.09.054
Yuen B (1991) Heuristics for sequencing cutting patterns. Eur J Oper Res 25(2):183–190. https://doi.org/10.1016/0377-2217(91)90222-H

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.