RAIRO-Oper. Res. 57 (2023) 183–200
https://doi.org/10.1051/ro/2023001

RAIRO Operations Research
www.rairo-ro.org

ARC-FLOW FORMULATIONS FOR THE ONE-DIMENSIONAL CUTTING STOCK PROBLEM WITH MULTIPLE MANUFACTURING MODES

Heloisa Vasques da Silva[1,*], Felipe Kesrouani Lemos[1], Adriana Cristina Cherri[1] and Silvio Alexandre de Araujo[2]

Abstract. In this paper, an integration of the one-dimensional cutting stock problem with an operational problem that arises in the manufacture of concrete poles is studied. Seeing that poles have a steel structure, different thicknesses of steel bars can be used in their manufacture. This variety in combining the materials to produce the structure of the poles is known as alternative production modes or multiple manufacturing modes. The problem considered here has the objective of minimizing the total cost to meet the demand for poles using the different available configurations. This problem has already been introduced in the literature and it has been formulated as an integer programming problem. To solve it, the column generation procedure was used. The contribution of this paper is to reformulate the cutting stock problem with multiple manufacturing modes using arc-flow formulations. Arc-flow formulations are promising tools to model and solve complex combinatorial problems. Computational tests are performed comparing the formulations using instances from the literature, which were generated based on the data from a civil construction plant. The arc-flow formulations increased the number of instances solved to proven optimality and also reduced solution time. Lower and upper bounds are also improved when compared with the solution proposed in the literature.

Mathematics Subject Classification. 90C10, 90C27.

Received March 10, 2022. Accepted December 26, 2022.

1. Introduction

The cutting stock problem (CSP) aims to minimize the number of objects with fixed dimensions to be cut into weakly heterogeneous demanded items [51]. The increasing interest in this problem can be explained by its applicability, diversity of real-world problems, and its combinatorial complexity [44]. Indeed, the technical and economic importance of CSP applications in industrial and logisticals processes is one of the main reasons for the development of this research. The CSP appears in several industrial processes, where items are ordered with specified dimensions and objects, generally available in stock, can be steel bars [31], paper rolls [24, 41], metal or wooden sheets [26], glass sheets [39], precast beams [3], among others.

Keywords. Arc-flow formulation, cutting stock problems, reflect, meet in the middle, mathematical model.

[1] São Paulo State University (UNESP), Avenida Eng. Luís Edmundo Carrijo Coube, 14-01, Vargem Limpa, Bauru-SP, Brasil-CEP 17033-360, Brazil.

[2] São Paulo State University (UNESP), Rua Cristóvão Colombo, 2265, Jardim Nazareth, São José do Rio Preto-SP, Brasil CEP 15054-000, Brazil.

*Corresponding author: vasques.helo@gmail.com

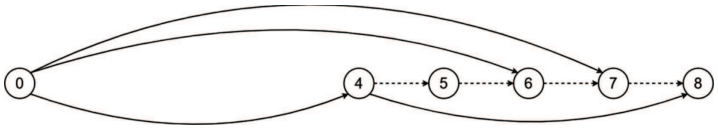


FIGURE 1. Arc-flow representation for a cutting stock problem.

Over the years, studies have focused on models and methods for the classical CSP [4, 7, 18, 21, 25, 34, 48, 50], alongside classifications and literature reviews [13, 19, 49, 51]. In industry, however, the CSP is not an isolated process and better contributes to the business goals if handled together with other processes [20]. Many researchers have dedicated their studies to the integration of the CSP with operational characteristics [53], like scheduling [5, 54], lot-sizing [23, 37], saw cycles [32], batch production [43] and alternative manufacturing modes [31]. Such operational characteristics are present in diverse productive environments, such as the concrete industry [31, 45], the mattress industry [8], the paper industry [41] and the automotive spring factory [2].

This paper focuses on the integration between the one-dimensional cutting stock problem and the alternative manufacturing modes problem (CSP-AM) proposed by Lemos *et al.* [31]. This problem concerns operational aspects found in the construction industry, in which there are alternative manufacturing modes to meet the same demand; that is, a final product can be produced with different combinations of items preserving the strength requirements of the structure. For the problem being studied, a particular concrete pole can be manufactured with different combinations of lengths and thicknesses of steel bars. In [31], this problem was modelled using cutting patterns (*i.e.*, possible combinations of items cut from the object, satisfying its length limit) and the column generation procedure with a heuristic was used as the solution method. The column generation procedure was proposed by Gilmore and Gomory [21, 22] and it is well-known for its high-quality continuous relaxation value. The number of variables in pattern-based formulations is exponential and depends on the number of items [12].

Another formulation for the CSP proposed in the literature is the well-known arc-flow formulation by [48]. The idea of flow to solve cutting and packing problems dates from [52] that solved a CSP as a network flow problem with side constraints. In the arc-flow formulation for the CSP proposed by Valério de Carvalho [48], the nodes are associated with positions into the object, and arcs, that connect the nodes, are associated with items or loss. A path from node 0 to node L (object length) characterizes a valid cutting pattern. Figure 1 illustrates an arc-flow network for an object of length $L = 8$, and items of length $l_i = (7, 6, 4)$, where arcs with straight lines represent a possible item allocation and arcs with dotted lines represent possible loss of material.

This formulation is pseudo-polynomial, *i.e.*, the number of variables and the number of constraints is polynomial in both number of items and object length [12] and has strong linear relaxation [35,48]. Recent developments in mixed integer linear programming (MILP) commercial software and increases in computational power have intensified interest in the application of this kind of formulation in different combinatorial problems [13].

The contribution of this paper to the literature is it proposes three different approaches based on arc-flow formulations for the cutting stock problem with multiple manufacturing modes [31]. These formulations are based on the previous papers of Valério de Carvalho [48], Côté and Iori [10], and Delorme and Iori [12]. Computational tests were carried out to evaluate the performance of the proposed formulations and results compared to those in the existing literature, showing the high quality of the proposed formulations.

The remainder of this paper is organized as follows: Section 2 reviews related studies in the literature. Section 3 describes the cutting stock problem with multiple manufacturing modes. Section 4 presents the proposed arc-flow formulations, as well as an illustrative example. Section 5 conducts the experimental tests and discusses the results. Section 6 concludes the paper and presents possible future research.

2. Literature review

In this section, characteristics and applications of arc-flow formulations are presented, which are powerful tools to model and solve complex combinatorial problems, providing good results [11]. The applications cover extensions for the cutting and packing problem [14, 33, 36, 40], vehicle routing [42, 46], and scheduling [15, 27–30, 47]. Recently, [11] published a wide discussion of arc-flow formulations, presenting many applications and methods based on arc-flow formulations and also described the relationship of the formulation with dynamic programming. The literature review that follows will focus on papers that apply arc-flow formulations to cutting stock problems.

Determining the arcs in the arc-flow formulations is important for their efficiency since arcs are related to the model variables and the active nodes are related to constraints. A network with a reduced number of arcs may lead to symmetry reduction, a strong linear relaxation and a smaller branch-and-bound tree [11]. Reduction techniques aiming to remove redundant arcs from the network without losing optimality have been studied by researchers, like the reduction criteria proposed by [48].

In [6] an exact solution method was proposed that builds a compact graph with a size reduced even further without dropping any path needed to determine an optimal solution for arc-flow formulations. Additionally, its linear relaxation is equivalent to [21, 22]. To reduce arc-flow formulation size, [9] studied a generic solution structure that utilizes state-space relaxation techniques and can be applied to a variety of problems. The method utilizes an iterative aggregation/disaggregation scheme that modifies the possibility of combining arcs in viable paths. The structure was evaluated in CSP and vehicle routing instances.

In [10] the *meet-in-the-middle* technique was proposed. In this technique, each flow representing a valid cutting pattern could be converted in an equivalent flow aligning items to the left and the right of a certain position in the object. This technique reduces the formulation size as it eliminates some nodes and arcs present initially, enabling its adaptation to several problems, such as vehicle routing and scheduling, including multidimensional cutting stock and bin packing problems. A pseudo-polynomial formulation to solve bin packing and CSP that uses only half the object length was proposed by [12]. With fewer nodes and, consequently, fewer arcs, this formulation has fewer variables and constraints than the classical arc-flow formulation. The authors presented strategies to improve the computational performance of the formulation using techniques based on column generation, cutting planes and heuristic procedures. The formulation was also extended to solve the variable-sized bin packing problem and the bin packing problem with item fragmentation.

For small instances, it is possible to solve arc-flow formulations directly using a MILP solver. However, depending on the problem parameters, a large number of variables and constraints could be generated, making it necessary to rely on other solution methods to solve the problem in a reasonable computational time. One can use, for instance, branch-and-bound algorithms [48], branch-and-price-and-cut algorithms [1] based on the column generation procedure, the Benders decomposition algorithm [14], or the Dantzig–Wolfe decomposition [28].

Recent research has shown that arc-flow formulations can be adapted to model many problems and their solution methods used to solve such problems. Besides, MILP software can solve some small instances until proven optimality. However, more complex instances, leading to more variables and constraints, demand more computational effort. Indeed, notable studies are dealing with reduction techniques, as well as theoretical development and algorithm improvements. Studies of this formulation have become attractive since it has the potential to develop solution procedures for several combinatorial problems.

3. Problem description

This section describes the one-dimensional cutting stock problem with alternative manufacturing modes proposed by [31]. This problem arises in the production process in the reinforced concrete industry; specifically, in the production process for concrete poles. The use of different steel bar thicknesses in the structure of the

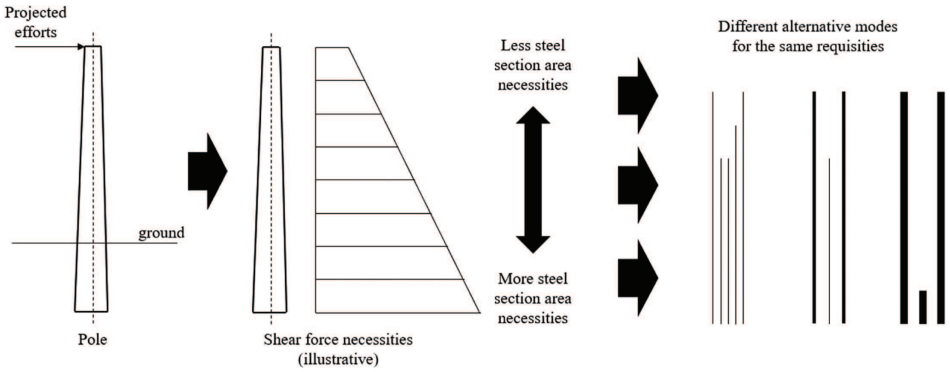


FIGURE 2. Alternative modes for a single product [31].

products allows a pole to have different combinations of steel bars, varying in lengths and thicknesses, without impacting its quality.

Figure 2 illustrates the problem: a certain force projected on the pole is represented as a shear force diagram. The multiple manufacturing modes reflect the possibility of different combinations of steel bars lengths and thicknesses meeting the same technical requirements.

This problem aims to meet the demand for all final products, in this case, poles, choosing the most convenient manufacturing modes to minimize the total raw material costs. Each option of the one-dimensional raw material has a cost depending on its thickness. Each production mode is characterized by a combination of a certain quantity of one-dimensional items with a stated thickness and length. Thus, the formulation aims to decide: (i) how many products will be produced using each alternative mode, and (ii) how steel bars will be cut to meet the demand of final products.

It seems trivial to choose one production mode for a single product considering only the raw material costs. However, analyzing a production mix with different products and demands, the combination possibilities increase. In this way, the overall cost is not always minimized by choosing the best mode for every single product, as steel leftovers may be combined to meet the total demand [31].

The formulation proposed by [31] for the one-dimensional cutting stock problem with alternative production modes (1D-CSP-AM) is presented in (1)–(5). The following sets, parameters, and decision variables are used:

Sets:

- J ($j \in J,\ j = 1, \ldots, \mathrm{NJ}$): set of final products;
- K ($k \in K,\ k = 1, \ldots, \mathrm{NK}$): set of thickness of steel bars;
- I ($i \in I,\ i = 1, \ldots, \mathrm{NI}$): set of item lengths;
- M_j ($m \in M_j$): set of alternative production modes to meet the demand for product j;
- P ($p \in P,\ p = 1, \ldots, \mathrm{NP}$): set of cutting patterns.

Parameters:

- L: length of steel bars;
- l_i: length of item i;
- d_j: demand for final product j;
- θ_k: cost of steel bars of thickness k;
- b_{ikm}: amount of item of length l_i and thickness k used in mode m;
- a_{ip}: amount of item of length l_i of cutting pattern p.

Decision variables:

- X_{pk}: amount of steel bars of thickness k cut according to pattern p;
- Z_{jm}: amount of final products j produced according to mode m.

$$\min \quad \sum_{p\in P}\sum_{k\in K}\theta_k X_{pk} \tag{1}$$

$$\text{s.t.}: \quad \sum_{m\in Mj} Z_{jm} \geq d_j, \qquad j \in J, \tag{2}$$

$$\sum_{p\in P} a_{ip}X_{pk} \geq \sum_{j\in J}\sum_{m\in M_j} b_{ikm}Z_{jm}, \quad k \in K,\ i \in I, \tag{3}$$

$$Z_{jm} \in \mathbb{Z}_+, \qquad j \in J,\ m \in M_j, \tag{4}$$

$$X_{pk} \in \mathbb{Z}_+, \qquad p \in P,\ k \in K. \tag{5}$$

In the model (1)–(5), the objective function (1) minimizes the raw materials total cost. Constraints (2) ensure that all the final product demands are met. Constraints (3) ensure that the demand for items is met according to alternative production modes. Constraints (4) and (5) define the decision variable domains.

The formulation proposed by [31] for the CSP-AM uses cutting patterns, which are the possible ways of cutting steel bars into items satisfying the bar length limit. To deal with the large number of variables due to multiple possible cutting patterns, [31] used the column generation method based on [21, 22] to generate columns for the optimal solution of the relaxed linear model. For this purpose, integrality conditions (4) and (5) were relaxed and the restricted master problem was initialized with homogeneous cutting patterns. At each iteration, the dual values referring to constraints (3) were used in a subproblem that minimizes the relative cost associated with a new set of variables and generates new columns. Subsequently, the columns generated were used in the formulation to obtain an integer solution with the CPLEX solver. It is, therefore, a heuristic method, as the integer solution is obtained using only the columns generated during the column generation process; that is, some columns that possibly would be part of the optimal solution were not available. This type of strategy has been widely used (for example, [16], [38] and [30]) with good results.

4. Arc-flow formulations

In this section, three different approaches are proposed to represent the CSP-AM aiming for optimal solutions using MILP solvers. The formulations are based on the *classical arc-flow* formulation proposed by Valério de Carvalho [48], the *meet-in-the-middle* principle proposed by Côté and Iori [10] and the *reflect* formulation proposed by Delorme and Iori [12].

At the end of this section, to compare the set of arcs and nodes of these approaches, an illustrative example is given.

4.1. Classical arc-flow formulation

In [48], the CSP was modelled as a problem to find a path in an acyclic directed graph with $L+1$ vertices. Formally, let $G = (V, A)$ be a graph with $V = \{0, 1, \ldots, L\}$ being a set of vertices where L is the length of the object. The distance from one vertex to the next represents one length unit. Let $A = \{(a, b) : a < b;\ b - a = l_i,\ i \in I;\ a, b \in V\} \cup \{(a, a+1) : \min(l_i) \leq a,\ a \in V\backslash\{L\}\}$ be a set of arcs representing (i) the cut of an item of length $b - a = l_i$, or (ii) a loss arc $(a, a+1)$. A possible cutting pattern corresponds to a path from root node 0 to the sink node L.

To model the CSP-AM, with different raw material thicknesses, the notation used is presented as follows. Let K $(k \in K,\ k = 1, \ldots, NK)$ be a set of thickness options of the one-dimensional raw material with cost θ_k

depending on thickness k. Let I ($i \in I,\ i = 1, \ldots, NI$) represent a set of item lengths cut from raw materials. J ($j \in J,\ j = 1, \ldots, NJ$) represents a set of final products with demand d_j. Set M_j ($m \in M_j$) represents alternative production modes associated to each product j. Each mode m is characterized by a combination of one-dimensional items with thickness k and length l_i in quantity b_{ikm}.

The decision variables F_k represent the number of steel bars of thickness k used, X_{abk} represents the number of times that steel bars of thickness k are cut in items of length $b - a$, and Z_{jm} represents the number of each product j manufactured according to mode m.

Decision variables:

- F_k: number of steel bars of thickness k used;
- X_{abk}: number of items of length $b - a$ cut from objects of thickness k;
- Z_{jm}: number of final products j produced according to mode m.

The classical arc-flow formulation adapted for the CSP-AM is presented in (6)–(14).

$$\min \quad \sum_{k \in K} \theta_k F_k \tag{6}$$

$$\text{s.t.:} \quad -\sum_{(0,b) \in A} X_{0bk} = -F_k, \qquad \forall\ k \in K, \tag{7}$$

$$\sum_{(a,b) \in A} X_{abk} - \sum_{(b,c) \in A} X_{bck} = 0, \qquad b = 1, \ldots, L-1,\ \ k \in K, \tag{8}$$

$$\sum_{(a,L) \in A} X_{aLk} = F_k, \qquad k \in K, \tag{9}$$

$$\sum_{m \in M_j} Z_{jm} \geq d_j, \qquad j \in J, \tag{10}$$

$$\sum_{(a,a+l_i) \in A} X_{a,a+l_i,k} \geq \sum_{j \in J} \sum_{m \in M_j} b_{ikm} Z_{jm}, \quad i \in I,\ k \in K, \tag{11}$$

$$X_{abk} \in \mathbb{Z}_+, \qquad (a,b) \in A,\ k \in K, \tag{12}$$

$$F_k \in \mathbb{Z}_+, \qquad k \in K, \tag{13}$$

$$Z_{jm} \in \mathbb{Z}_+, \qquad j \in J,\ m \in M_j. \tag{14}$$

The objective function (6) minimizes the total cost of raw materials. Constraints (7)–(9) are derived from the formulation of [48] and impose flow conservation, that is, all flow entering a vertex must exit, and the flow in the graph remains the same, hence the flow leaving node 0 is the same as that entering node L. Constraints (10) ensure that all the product demands are met. Constraints (11) ensure that demand for items is met according to production modes. Constraints (12)–(14) define the decision variable domains. The computational behaviour of this formulation depends on the arc set that must be as small as possible and ensure optimization [10].

To eliminate redundant arcs from the network, two reduction criteria proposed by [48] were used. The first criterion consists of allocating items according to decreasing length, that is, an item of length l_1 can only be allocated after an item l_2 if $l_1 \leq l_2$, or at the beginning of the object. The second criterion imposes that loss arcs cannot be allocated before item arcs. Although there is a third criterion, related to the demand, it was not applicable for the CSP-AM.

4.2. Meet-in-the-middle approach

To obtain even smaller networks, reducing the number of arcs and, consequently, variables from the formulation, [10] proposed the *meet-in-the-middle* principle. The idea of this strategy is to take advantage of the order of the items to obtain a smaller set of arcs. According to [10], cutting items according to a given order (the one

usually adopted is that of non-increasing length) preserves optimality and, consequently, the set of positions where an item can be cut may be replaced by a smaller set, or at least an equivalent set, that considers such order.

To do so, one can fix a certain threshold value (t) along the steel bar length and align all items whose left corner a is at the left of t ($a \leq t-1$) to the left of the steel bar and all other items ($a \geq t$) to the right corner of the steel bar. In this way, each path representing a cutting pattern can be transformed into an equivalent one. The chosen threshold t is the one that minimizes the total patterns and is used to build the arc set.

Assuming items are sorted in non-increasing order and that, for any item $i = 1, \ldots, \mathrm{NI}$, the maximum number of items i cut from the object is $\max_i = \lfloor l_i/L \rfloor$, and $\max_j^i = \max_j$ for any $j = 1, \ldots, i-1$ and $\max_i^i = \max_i - 1$. The left and right patterns for each item i and each threshold t are defined by (15) and (16).

$$L'_{it} = \left\{ a = \sum_{j=1}^{i} l_j \xi_j : 0 \leq a \leq \min\{t-1, L-l_i\}, \quad \xi_j \in \left\{0, 1, \ldots, \max_j^i\right\}, \quad \text{for} \quad j = 1, 2, \ldots, i \right\}, \tag{15}$$

$$R'_{it} = \left\{ L - l_i - a : a = \sum_{j=1}^{i} l_j \xi_j, \quad 0 \leq a \leq L - l_i - t, \ \xi_j \in \left\{0, 1, \ldots, \max_j^i\right\}, \quad \text{for} \quad j = 1, 2, \ldots, i \right\}. \tag{16}$$

The set of meet-in-the-middle patterns is the union of left and right patterns, defined by (17) and (18).

$$M'_{it} = L'_{it} \cup R'_{it} \tag{17}$$

$$M'_t = \bigcup_{i \in I} M'_{it} \tag{18}$$

$$M' = \left\{ M'_t : t = \underset{s \in \{1,\ldots,L\}}{\arg\min} \sum_{i \in I} |M'_{is}| \right\}. \tag{19}$$

The minimal set of meet-in-the-middle patterns (M') is determined by (19), assuming the threshold t that minimizes the total number of patterns.

To build the arc set using the *meet-in-the-middle* principle, the same two criteria proposed by [48] and explained in Section 4.1 are used. Three criteria proposed by [10] are also included in this formulation:

- Remove the original unit-length loss arcs $(a, a+1)$, and introduce only loss arcs connecting a pair of consecutive vertices in the reduced set of patterns;
- Remove a loss arc connecting two vertices if there is an item arc connecting the same two vertices;
- Impose that items having length larger than $L/2$ are only cut with their lowest corner in 0 (because no larger item can be cut from the same object and the non-increasing length order must be fulfilled).

The arc set built using this strategy is then used in the classical arc-flow formulation CSP-AM defined by (6)–(14).

4.3. Reflect formulation

A pseudo-polynomial formulation called *reflect* was proposed by [12], which requires only half of the object length to be used in the network. The formulation is built on a multigraph $G = (V, A)$, where $V = \{0, 1, \ldots, L/2\}$ is a set of all partial lengths of the object. Reflect strategy considers almost the same arcs as in the classical arc-flow formulation but removes all arcs starting in or after $L/2$. Also, it "reflects" item arcs (d, e) having $d < L/2$ and $e > L/2$ into an arc $(d, L-e)$. This strategy creates a significantly smaller formulation compared to the classical arc-flow formulation.

Formally, the set of arcs A is divided into A_s and A_r. A_s represents the set of *standard arcs*, whose arcs represent items or loss that starts and ends before or in $L/2$, *i.e.*, proceeds from left to right. A_r represents the set of *reflected arcs*, *i.e.*, arcs representing items that pass through the *reflect* vertex.

A generic arc belonging to either A_s or A_r is represented by (d, e, g), in which d and e are the nodes with $e > d$, and g can be s or r representing a standard or reflected arc. Therefore, an arc belonging to A_s is denoted as (d, e, s), and an arc belonging to A_r is denoted as (d, e, r). To define a subset of arcs associated with an item of length i $A_i = \{(d, d + l_i, s) \in A_s \cup (d, L - (d + l_i), r) \in A_r\}$ is used. $\delta_s^-(e)$ and $\delta_r^-(e)$ are used to denote standard or reflected arcs entering the vertex e and $\delta_s^+(e)$ and $\delta_r^+(e)$ are used to denote standard and reflected arcs leaving the vertex e.

To formulate CSP-AM using the reflect formulation, the sets and parameters already defined in Section 4.1 are used. The decision variables are defined by:

- ξ_{degk}: number of times that arc (d, e, g) is chosen for the raw material of thickness k;
- Z_{jm}: number of each final product j produced according to mode m.

The CSP-AM based on the reflect formulation [12] is presented in (20)–(26).

$$\min \quad \sum_{k \in K} \sum_{(d,e,r) \in A_r} \theta_k \xi_{derk} \tag{20}$$

$$\text{s.t.:} \quad \sum_{(d,e,s) \in \delta_s^-(e)} \xi_{desk} = \sum_{(d,e,r) \in \delta_r^-(e)} \xi_{derk} + \sum_{(e,f,g) \in \delta^+(e)} \xi_{efgk}, \quad k \in K,\ e \in V \backslash \{0\}, \tag{21}$$

$$\sum_{(0,e,g) \in \delta^+(0)} \xi_{0egk} = 2 \sum_{(d,e,r) \in A_r} \xi_{derk}, \qquad k \in K, \tag{22}$$

$$\sum_{m \in M_j} Z_{jm} \geq d_j, \qquad j \in J, \tag{23}$$

$$\sum_{(d,e,g) \in A_i} \xi_{degk} \geq \sum_{j \in J} \sum_{m \in M_j} b_{ikm} Z_{jm}, \qquad i \in I, \quad k \in K, \tag{24}$$

$$\xi_{degk} \in \mathbb{N}, \qquad (d, e, g) \in A, \quad k \in K, \tag{25}$$

$$Z_{jm} \in \mathbb{Z}_+, \qquad j \in J, \quad m \in M_j. \tag{26}$$

The objective function (20) minimizes the raw material total cost. Constraints (21) impose that the amount of flow entering a node e in standard arcs is equal to the amount of flow (in both standard and reflected arcs) leaving node e plus the amount of flow of the reflected arcs entering into e for each raw material thickness k. Constraints (22) impose boundary conditions by forcing the amount of flow leaving 0 to be twice the number of objects used. Constraints (23) assure that all product demand is met. Constraints (24) assure that item demands are met according to production modes. Constraints (25) and (26) define the decision variable domains.

4.4. Illustrative example

This illustrative example is taken from [31] and is used here to show the differences among the sets of vertices and arcs using the three approaches. The steel bars have length $L = 10$ m. There are 4 thicknesses of raw materials (NK = 4), and the costs are $\theta_k = (40, 25, 20, 10)$. Two products (NJ = 2) must be produced with final demand of 100 units each ($d_1 = d_2 = 100$). There are 4 alternatives modes ($M_1 = M_2 = 4$) to manufacture each product. Items should be cut in 4 different lengths ($l_i = (7, 6, 4, 3)$). Table 1 shows all combinations of item lengths and thicknesses in each production mode.

TABLE 1. Alternative modes for each product to be manufactured [31].

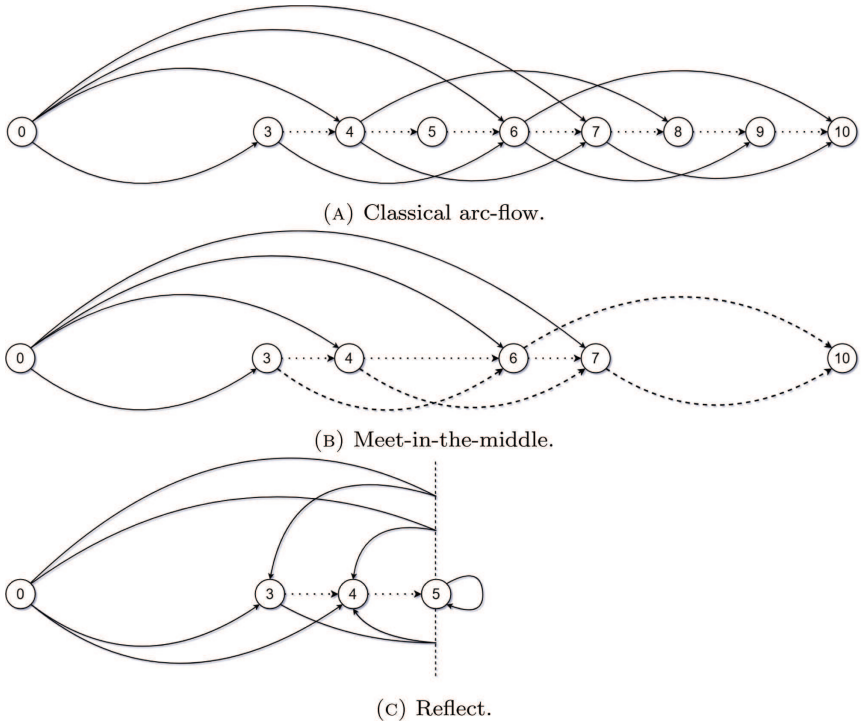
Products (j)	Manufacturing mode (m)	Raw material (k)	Item length (l_i)	Required quantity (b_{ikm})
1	1	1	7 m	4
		2	3 m	2
		3	–	–
		4	–	–
	2	1	–	–
		2	7 m	6
		3	4 m	2
		4	–	–
	3	1	–	–
		2	–	–
		3	7 m	6
		4	6 m	8
	4	1	7 m	4
		2	–	–
		3	–	–
		4	7 m	2
2	1	1	7 m	4
		2	6 m	2
		3	–	–
		4	–	–
	2	1	–	–
		2	4 m	8
		3	6 m	6
		4	–	–
	3	1	–	–
		2	–	–
		3	3 m	16
		4	4 m	12
	4	1	3 m	8
		2	–	–
		3	–	–
		4	7 m	8

Figure 3 shows the arc sets built for the classical arc-flow formulation (3a), meet-in-the-middle approach (3b) and reflect formulation (3c). Each vertex represents a unit length of the steel bar. Arcs represent items or losses. Loss arcs are represented with dotted lines.

In the meet-in-the-middle approach, the chosen threshold was $t = 3$, so arcs beginning in vertices $a \geq 3$ are right-aligned patterns represented with dashed lines, while arcs beginning in vertices $a \leq 2$ are left-aligned patterns represented with straight lines.

In the reflect formulation, where only half of the bar length is used, the reflected arcs pass through the vertex $L/2 = 5$. Note that there can be a standard arc and a reflected arc with the same initial and end node, as in $(0, 3, s)$ and $(0, 3, r)$ in Figure 3c, where the first represents a standard arc associated with an item of length 3, and the second represents a reflected arc associated with an item of length 7. The arc $(L/2, L/2, r)$ is included in the reflected arcs to denote possible paths meeting in $L/2$.

The classical arc-flow formulation shows 9 vertices and 17 arcs, which is reduced to 6 vertices and 11 arcs with the meet-in-the-middle approach. The reflect formulation has 4 vertices and 8 arcs.



(A) Classical arc-flow.

(B) Meet-in-the-middle.

(C) Reflect.

FIGURE 3. Example of arcs for different arc-flow formulations. (A) Classical arc-flow, (B) Meet-in-the-middle and (C) Reflect.

5. Computational results and discussion

In this section, a set of 1215 random instances were tested to evaluate the formulation performances and bounds. All the experiments were executed on a 16 GB RAM computer with a 2.2 GHz hexacore Intel Core i7-8750H processor, using CPLEX (version 12.10) as solver and Optimization Programming Language (OPL) used to build the mathematical models. Visual Basic for Applications was used to handle the data, generate the files to be processed by the OPL, as well as to analyze the results.

5.1. Data set

The 1215 instances used were those proposed in [31]. In total, 30 final products (NJ = 30) should be produced. The length of the objects was $L = 1200$ and there were 20, 30 and 40 different items (NI). The number of thicknesses (NK) had three scenarios: 2, 5 and 8. The number of different alternative modes per product (M_j) also had three scenarios: 5, 10 and 15. The item lengths (l_i) were generated either "small" ($l_i \in U(120,\ 360)$), "large" ($l_i \in U(360,\ 840)$) and "varied" ($l_i \in U(120,\ 840)$). Objects costs were considered either "identical" ($\theta_k = 1$), "homogeneous" ($\theta_k \in U(1,\ 10)$) and "heterogeneous" ($\theta_k \in U(1,\ 100)$). The notation U here represents a uniform distribution in a discrete set.

Demands for final products were always generated between 1 and 10 units ($d_j \in U(1,\ 10)$). Quantities required of an item i of a certain thickness k in an alternative mode m (b_{ikm}) were also generated randomly, considering a 98% chance of being null ($b_{ikm} = 0$) and a 2% chance of following a uniform distribution between 2 and 8 units ($b_{ikm} \in U(2,\ 8)$). The threshold of 98% was inspired by the industrial example of [31], where 98% of elements b_{ikm} were equal to 0. The combination of these scenarios generated 243 sets of parameters, for which 5 different instances were generated randomly per set, making 1215 instances altogether.

TABLE 2. Average time (average column or arc generation time) for each formulation when varying parameters.

		CG	AF	MIM	RE
Items	Small	319 (199.6)	145 (1.00)	144 (5.34)	**137 (0.04)**
	Large	29 (26.7)	25 (0.01)	10 (0.01)	**4 (0.00)**
	Varied	135 (69.8)	121 (0.10)	113 (0.76)	**110 (0.00)**
NK	2	44 (13.7)	49 (0.36)	43 (2.02)	**39 (0.01)**
	5	135 (74.9)	107 (0.37)	101 (2.02)	**95 (0.01)**
	8	305 (207.6)	135 (0.38)	124 (2.07)	**116 (0.01)**
NM	5	148 (79.4)	100 (0.36)	93 (1.94)	**89 (0.01)**
	10	163 (100.2)	97 (0.36)	89 (2.07)	**83 (0.01)**
	15	173 (116.5)	94 (0.38)	85 (2.10)	**79 (0.01)**
NI	20	**69** (29.4)	82 (0.09)	75 (0.68)	72 **(0.00)**
	30	129 (63.1)	102 (0.32)	92 (1.86)	**85 (0.01)**
	40	285 (203.7)	108 (0.70)	100 (3.56)	**94 (0.02)**
Costs	Homogeneous	157 (95.1)	100 (0.36)	90 (2.00)	**84 (0.01)**
	Heterogeneous	184 (100.3)	119 (0.36)	111 (2.01)	**106 (0.01)**
	Identical	142 (100.8)	72 (0.38)	65 (2.10)	**61 (0.01)**
Overall		161 (98.7)	97 (0.37)	89 (2.04)	**84 (0.01)**

5.2. Results

For the proposed formulations to the CSP-AM problem, AF is used for the one based on the classical Arc-Flow formulation, MIM for the one based on the Meet-In-the-Middle principle, and RE for the one based on the reflect formulation. CG is used for the solution method based on a Column Generation and a heuristic procedure proposed by [31]. Data were subjected to each of the different formulations. The column generation and the arc generation ran without time limitation. To find the integer solution, the time was limited to 180 s and all instances reached an integer solution within the time limit. The solution time considered in the analysis is the time taken to generate columns or arcs added to the time to find the integer solution, so it can exceed 180 s.

Table 2 shows the average time for the formulations and the average time for generating columns or arcs as the parameters vary. On average, column generation takes about 75.1% of the computational time used, while arc generation in AF, MIM and RE takes up 2.9%, 10.6% and 0.6%, respectively. Bold values highlight the best average time among the approaches.

Figure 4 shows the performance profile proposed by [17]. For each instance, a normalized time τ was computed as the ratio of the run time over the minimum run time for solving the instance. For each value of τ, the vertical axis shows the percentage of instances for which the corresponding formulation spent at most τ times the computing time of the fastest formulation. The curves start from the percentage of instances in which the formulation was the fastest: 14%, 2%, 7%, and 76% respectively for the CG, AF, MIM, and RE. The right end of the chart shows the percentage of instances that reached an integer solution within the time limit, which is 100% for all methods. Figure 4 graphically demonstrates that RE is the most efficient formulation for these instances, as its curve is in the upper part of the chart.

Table 3 summarizes the main results of running the formulations for the 1215 instances. Gaps were calculated using Upper Bounds (UB) and Lower Bounds (LB) in the exact formulations with the expression by (UB − LB)/LB. For CG approach, the relaxed solution of the formulation after the column generation procedure was used as a valid lower bound.

As the CG approach uses a heuristic method to obtain the integer solution, an optimal solution is proven when the relaxed solution and the upper bound are equal. RE formulation presented more instances with proven

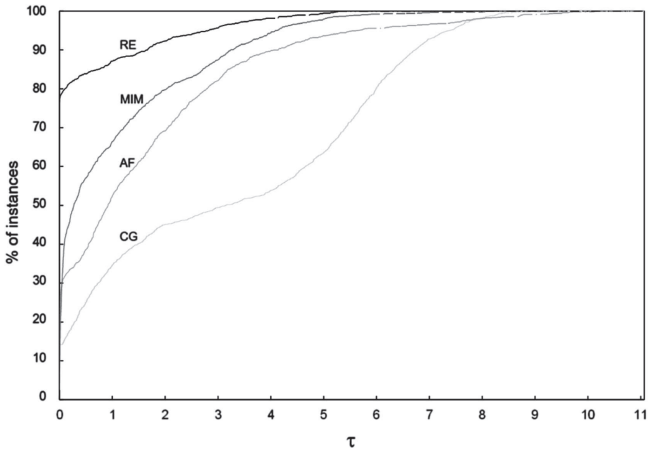


FIGURE 4. Performance profile of the formulations.

TABLE 3. Overall performance of the formulations.

Overall (1215 instances)	CG	AF	MIM	RE
Average gap (%)	0.59	0.32	0.27	0.25
Average time (s)	161.1	97.1	89.1	83.8
Average col./arc generation compared to total time (%)	75.1	2.9	10.6	0.6
Number of proven optimal instances	30	551	609	613
Average LB's improvement compared with CG (%)	–	0.07	0.10	0.11
Average UB's improvement compared with CG (%)	–	0.20	0.22	0.23

optimal solutions. Arc-flow formulations showed, on average, better gaps, times and bounds compared with CG approach.

In the following sections, the results analysis is separated into instances solved to proven optimality and non-proven optimality. At the end of the section, an analysis to evaluate the influence of the parameters in the results is presented.

5.2.1. Proven optimal instances

Table 4 gives the number of instances with proven optimal solutions found by each model and the average time of these instances separated by parameters. The best value for each parameter variation is highlighted in this table and in Tables 5–8. In general, instances with large items presented more instances solved in the optimality as well as shortest computational time. Table 4 also shows that RE solved more instances with the shortest average running time. Arc-flow formulations obtained proven optimal solutions for about half of the instances, while the CG approach obtained the proven optimal solution for only 30 instances.

In total, 646 of the 1215 instances obtained optimal solution in at least one of the formulations. In Table 5, the solution time of these instances in AF, MIM and RE were compared with the solution time in the CG approach, even if the solution was not proven optimal in the CG approach. The comparison shows that all three arc-flow-based formulations reach more optimal solutions with an average shorter running time. Indeed, the running time decreased, on average, by 56.6%, 68.5% and 79.7% in AF, MIM and RE, compared to CG. The biggest reductions are in instances with large items.

TABLE 4. Number of instances solved to proven optimality (average time in seconds) for each formulation when varying parameters.

		#inst	CG	AF	MIM	RE
Items	Small	405	0 (–)	96 (22.7)	106 (25.6)	**108 (16.5)**
	Large	405	30 (7.4)	310 (2.4)	**346** (2.7)	345 **(1.0)**
	Varied	405	0 (–)	145 (13.9)	157 (7.4)	**160 (7.3)**
NK	2	405	21 (4.4)	306 (7.8)	319 (6.8)	**320 (3.8)**
	5	405	9 (14.6)	156 (10.8)	178 (9.5)	**180 (8.3)**
	8	405	0 (–)	89 (10.0)	112 (8.7)	**113 (5.3)**
NM	5	405	13 (7.6)	172 (7.6)	**194** (7.8)	190 **(3.7)**
	10	405	11 **(4.8)**	185 (6.8)	202 (7.6)	**205** (5.3)
	15	405	6 (11.9)	194 (12.3)	213 (8.4)	**218 (6.9)**
NI	20	405	15 **(4.6)**	217 (6.8)	237 (5.1)	**240**(5.6)
	30	405	12 (6.2)	178 (12.1)	195 (9.2)	**198 (3.6)**
	40	405	3 (26.1)	156 (8.6)	**177** (10.4)	175 **(7.0)**
Costs	Homogeneous	405	9 (12.3)	195 (12.7)	216 (9.6)	**222 (6.9)**
	Heterogeneous	405	9 (5.3)	100 (5.3)	**119** (3.6)	114 **(2.8)**
	Identical	405	12 (5.3)	256 (7.6)	274 (8.5)	**277 (5.2)**
Overall		1215	30 (7.4)	551 (9.0)	609 (7.9)	**613 (5.4)**
Overall (% instances)			2.5	45.3	50.1	**50.5**

TABLE 5. Percentage of time reduction in instances solved to proven optimality for each formulation compared with CG when varying parameters.

		AF	MIM	RE
Items	Small	2.5	11.0	**57.0**
	Large	85.1	88.0	**95.9**
	Varied	31.6	**64.0**	60.1
NK	2	51.9	60.1	**86.0**
	5	61.0	**81.8**	66.0
	8	65.2	71.1	**83.8**
NM	5	58.4	66.7	**89.1**
	10	63.8	69.1	**85.9**
	15	48.2	**69.4**	65.6
NI	20	62.4	**74.7**	71.0
	30	31.3	56.1	**86.5**
	40	77.5	73.7	**83.9**
Costs	Homogeneous	76.3	77.2	**85.8**
	Heterogeneous	34.5	51.6	**74.3**
	Identical	49.5	**78.9**	75.5
Average		56.6	68.5	**79.7**

5.2.2. Non-proven optimal instances

Table 6 divides instances by parameter and shows the number of instances with non-proven optimal solution and their gaps in each approach. In total, the optimality was not proved for 569 instances. In these instances, approaches were compared by their upper and lower bounds. Instances with large items presented lower gaps. Also, the tightest average gaps were obtained in the RE formulation.

TABLE 6. Number of instances with non-proven optimality (average gap) for each formulation when varying parameters.

		#inst	CG	AF	MIM	RE
Items	Small	405	405 (1.02)	309 (0.96)	299 (0.85)	**297 (0.80)**
	Large	405	375 (0.26)	95 (0.06)	**59** (0.03)	60 **(0.01)**
	Varied	405	405 (0.51)	260 (0.33)	248 **(0.28)**	**245 (0.28)**
NK	2	405	384 (0.49)	99 (0.40)	86 **(0.36)**	**85** (0.37)
	5	405	396 (0.64)	249 (0.57)	227 (0.53)	**225 (0.49)**
	8	405	405 (0.68)	316 (0.66)	293 (0.59)	**292 (0.56)**
NM	5	405	392 (0.48)	233 (0.46)	**211** (0.45)	215 **(0.42)**
	10	405	394 (0.63)	220 (0.61)	203 (0.55)	**200 (0.52)**
	15	405	399 (0.70)	211 (0.69)	192 (0.62)	**187 (0.60)**
NI	20	405	390 (0.73)	188 (0.69)	168 (0.62)	**165 (0.61)**
	30	405	393 (0.62)	227 (0.59)	210 (0.55)	**207 (0.53)**
	40	405	402 (0.47)	249 (0.49)	**228** (0.46)	230 **(0.42)**
Costs	Homogeneous	405	396 (0.57)	210 (0.60)	189 (0.57)	**183 (0.54)**
	Heterogeneous	405	396 (0.61)	305 (0.46)	**286** (0.41)	291 **(0.39)**
	Identical	405	393 (0.64)	149 (0.81)	131 (0.76)	**128 (0.74)**
Overall		1215	1185 (0.61)	664 (0.58)	606 (0.54)	**602 (0.51)**

TABLE 7. Average percentage of upper bound improvement in instances with non-proven optimality compared with CG when varying parameters.

		AF	MIM	RE
Items	Small	0.06	0.14	**0.18**
	Large	**0.03**	**0.03**	**0.03**
	Varied	0.06	**0.09**	**0.09**
NK	2	0.13	0.14	**0.15**
	5	0.08	0.12	**0.15**
	8	0.02	0.10	**0.12**
NM	5	0.06	0.09	**0.11**
	10	0.04	0.10	**0.12**
	15	0.08	0.14	**0.17**
NI	20	0.13	0.20	**0.21**
	30	0.06	0.11	**0.13**
	40	0.01	0.05	**0.08**
Costs	Homogeneous	0.00	0.03	**0.07**
	Heterogeneous	0.04	0.11	**0.14**
	Identical	0.10	0.14	**0.15**
Average		0.06	0.11	**0.13**

Table 7 shows the percentage of improvement in the lower bounds of the AF, MIM and RE formulations compared to the CG approach separated by parameters. Instances with small items and with less items presented the biggest improvements. Overall, the RE formulation was, on average, the one that best managed to minimize the value of the solution.

For the lower bound case, shown in Table 8, instances with few items presented the highest percentage of improvement in the formulations, and, in general, the RE formulation was slightly superior to the others.

TABLE 8. Average percentage of lower bound improvement in instances with non-proven optimality compared with CG when varying parameters.

		AF	MIM	RE
Items	Small	0.08	0.11	**0.13**
	Large	0.16	0.17	**0.18**
	Varied	0.13	**0.16**	**0.16**
NK	2	0.09	**0.13**	**0.13**
	5	0.12	0.14	**0.15**
	8	0.10	0.13	**0.14**
NM	5	0.08	**0.10**	**0.10**
	10	0.11	0.14	**0.15**
	15	0.13	0.16	**0.17**
NI	20	0.21	0.25	**0.27**
	30	0.09	0.11	**0.12**
	40	0.04	0.06	**0.07**
Costs	Homogeneous	0.05	0.06	**0.07**
	Heterogeneous	0.11	0.12	**0.14**
	Identical	0.13	**0.17**	**0.17**
Average		0.11	0.13	**0.14**

5.3. Additional discussion

In the general comparison between column and arc generation, Table 2 shows that the column generation time is much higher than the arc generation time. The longest times required for column generation occur in instances with more bar thicknesses (NK), greater numbers of items (NI) and small-sized items (l_i). The number of thicknesses influences the column generation time because, according to the procedure used by [31], the cutting patterns are generated separately for each thickness k. The presence of more items generates more variables to solve the subproblem and the presence of smaller items allows more length combinations in the object, providing more cutting patterns to be tested and added to the formulation, as new columns are generated while the relative cost is attractive. In most instances (678 out of 1215), column generation takes 90%–100% of the total time. On average, for the 1215 instances, the column generation procedure takes 75.1% of the total time.

The arc generation for the arc-flow formulations depends on the quantity (NI) and the size of the items (l_i). Arcs are generated for each item and the number of vertices explored depends on its size. Therefore, having more and smaller items increases the arc generation time. It is worth mentioning that the length of the bar influences the generation time of the arcs. However, for the CSP-AM this was not a factor as the bars had the same length ($L = 1200$) in all instances. In the MIM approach, the arc generation took longer due to the steps of generating normal patterns, generating meet-in-the-middle patterns and finding the threshold (t) that minimizes the number of patterns. The generation of arcs in the RE formulation is the fastest since it only covers half of the positions in the object ($L/2$).

According to the comparisons presented in the Tables 2, 4 and 6, the outputs measured from random experiments of the formulations (times, gaps and bounds) are also influenced mainly by the item sizes (l_i), number of items (NI) and number of raw material thicknesses (NK). The variation of these parameters changes the number of variables and constraints for the formulations, influencing the results obtained. The presence of more thicknesses of bars and more items increases the number of variables and constraints in the formulations, demanding more effort to solve them. As for the size of the items, the presence of smaller items allows for more combinations and, therefore, more cutting patterns in the column generation approach and more cutting positions in the flow formulations. In the latter, each item and each possible cutting position in the different bars is associated with

an arc, thus generating different variables. The different positions of items in bars, in addition to generating arcs, generate active vertices. Each active vertex imposes new flow conservation constraints.

The instances with proven optimal solutions in arc-flow formulations have an advantage over the column generation approach in instances with large items because the existence of large items in the column generation approach leads to greater losses. Thus, the distance from the integer solution to the linear relaxation will be greater, requiring longer times to converge to the optimum in the restricted model. For arc-flow formulations, on the contrary, the presence of large items decreases the arc generation time and the number of variables and constraints, converging faster to the optimal solution.

In instances with non-proven optimal, the existence of fewer items and thicknesses is advantageous for the arc-flow formulations, as the smaller number of variables and constraints increases the speed of convergence, reaching better bounds.

6. Conclusion and future research

In this paper, the one-dimensional cutting stock problem with alternative manufacturing modes found in the literature is studied. To solve this problem, three approaches based on arc-flow formulations are proposed. The new approaches are based on the classical arc-flow formulation from [48] (AF), on the meet-in-the-middle principle from [10] (MIM), and on the reflect formulation from [12] (RE). Computational tests were performed comparing the results of these formulations with the one from [31], who proposed the integration of the cutting stock problem with multiple manufacturing modes developing a solution method based on column generation.

A total of 1215 instances were used based on an industrial application from [31] to compare the formulations. The analysis was divided into instances solved to proven optimality and instances with non-proven optimality.

Related to instances solved to proven optimality, arc-flow formulations (RE, MIM and AF) reduced the solution time on average by, 79.7%, 68.5%, and 56.6% respectively, compared with the approach previously described in the literature. Also, the arc-flow formulations increased the number of instances solved to proven optimality to about 50%, whereas in the approach based on column generation this value was about 2% of the instances. These instances are mostly those with large items and with few raw material thicknesses.

For instances with non-proven optimal solutions, RE, MIM and AF improved the quality of the solutions by 0.13%, 0.11% and 0.06%, respectively, on average. Furthermore, lower bounds were also improved on average by 0.14%, 0.13% and 0.11%, respectively, compared with the literature.

This study shows that there is room for future research. Other methods can be applied to the cutting stock problem with alternative manufacturing modes to solve instances that did not achieve a proven optimal solution. Additionally, arc-flow formulations can be applied to other extensions and integrations of the cutting stock problem, as they can improve the quality of the results reported so far in the literature.

Acknowledgements. The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP) for the support on this work. This research was funded by CNPq (process numbers 421130/2018-0, 305261/2018-5, 306558/2021-8 and 406335/2018-4) and FAPESP (process number 2013/07375-0 and 2016/01860-1).

References

[1] C. Alves and J.M. Valério de Carvalho, A branch-and-price-and-cut algorithm for the pattern minimization problem. *RAIRO: Oper. Res.* **42** (2008) 435–453.

[2] P.R.d.L. Andrade, S.A. de Araujo, A.C. Cherri and F.K. Lemos, The integrated lot sizing and cutting stock problem in an automotive spring factory. *Appl. Math. Model.* **91** (2021) 1023–1036.

[3] K.A.G.d. Araújo, T.d.O.E. Bonates and B.d.A. Prata, The integrated cutting and packing heterogeneous precast beams multiperiod production planning problem. *RAIRO: Oper. Res.* **55** (2021) 2491–2524.

[4] G. Belov and G. Scheithauer, A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *Eur. J. Oper. Res.* **171** (2006) 85–106.

[5] N. Braga, C. Alves, R. Macedo and J.M. Valério de Carvalho, Combined cutting stock and scheduling: A matheuristic approach. *Int. J. Innov. Comput. Appl.* **7** (2016) 135–146.

[6] F. Brandao and J.P. Pedroso, Bin packing and related problems: General arc-flow formulation with graph compression. *Comput. Oper. Res.* **69** (2016) 56–67.

[7] H. Cambazard and B. O'Sullivan, Propagating the bin packing constraint using linear programming, in *International Conference on Principles and Practice of Constraint Programming*, Springer (2010) 129–136.

[8] M.M. Christofoletti, S.A. de Araujo and A.C. Cherri, Integrated lot-sizing and cutting stock problem applied to the mattress industry. *J. Oper. Res. Soc.* **72** (2021) 1279–1293.

[9] F. Clautiaux, S. Hanafi, R. Macedo, M.-E. Voge and C. Alves, Iterative aggregation and disaggregation algorithm for pseudo-polynomial network flow models with side constraints. Eur. J. Oper. Res. **258** (2017) 467–477.

[10] J.F. Côté and M. Iori, The meet-in-the-middle principle for cutting and packing problems. *INFORMS J. Comput.* **30** (2018) 646–661.

[11] V.L. de Lima, C. Alves, F. Clautiaux, M. Iori and J.M. Valério de Carvalho, Arc flow formulations based on dynamic programming: Theoretical foundations and applications, *Eur. J. Oper. Res.* **296** (2022) 3–21.

[12] M. Delorme and M. Iori, Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems. *INFORMS J. Comput.* **32** (2020) 101–119.

[13] M. Delorme, M. Iori and S. Martello, Bin packing and cutting stock problems: Mathematical models and exact algorithms. *Eur. J. Oper. Res.* **255** (2016) 1–20.

[14] M. Delorme, M. Iori and S. Martello, Logic based Benders' decomposition for orthogonal stock cutting problems. *Comput. Oper. Res.* **78** (2017) 290–298.

[15] M. Delorme, M. Iori and N.F. Mendes, Solution methods for scheduling problems with sequence-dependent deterioration and maintenance events. *Eur. J. Oper. Res.* **295** (2021) 823–837.

[16] D.N. do Nascimento, S.A. de Araujo and A.C. Cherri, Integrated lot-sizing and one-dimensional cutting stock problem with usable leftovers. *Ann. Oper. Res.* **316** (2022) 785–803.

[17] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Math. Prog.* **91** (2002) 20 1–213.

[18] H. Dyckhoff, A new linear programming approach to the cutting stock problem. *Oper. Res.* **29** (1981) 1092–1104.

[19] H. Dyckhoff, A typology of cutting and packing problems. *Eur. J. Oper. Res.* **44** (1990) 145–159. DOI: 10.1016/0377-2217(90)90350-K.

[20] J. Erjavec, M. Gradišar and P. Trkman, Renovation of the cutting stock process. *Int. J. Prod. Res.* **47** (2009) 3979–3996.

[21] P.C. Gilmore and R.E. Gomory, A linear programming approach to the cutting-stock problem. *Oper. Res.* **9** (1961) 849–859.

[22] P.C. Gilmore and R.E. Gomory, A linear programming approach to the cutting stock problem – part ii. *Oper. Res.* **11** (1963) 863–888.

[23] N. Goulart, T.F. Noronha, M.G. Ravetti and M.C. de Souza, The integrated uncapacitated lot sizing and bin packing problem. *RAIRO: Oper. Res.* **55** (2021) 1197–1212.

[24] J. Kallrath, S. Rebennack, J. Kallrath and R. Kusche, Solving real-world cutting stock-problems in the paper industry: Mathematical approaches, experience and challenges. *Eur. J. Oper. Res.* **238** (2014) 374–389.

[25] L.V. Kantorovich, Mathematical methods of organizing and planning production. *Manag. Sci.*, **6** (1960) 366–422.

[26] E.S. Kokten and Ç. Sel, A cutting stock problem in the wood products industry: A two-stage solution approach. *Int. Trans. Oper. Res.* (2020). DOI: 10.1111/itor.12802.

[27] A. Kramer, M. Dell'Amico and M. Iori, Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. *Eur. J. Oper. Res.* **275** (2019) 67–79.

[28] A. Kramer, M. Dell'Amico, D. Feillet and M. Iori, Scheduling jobs with release dates on identical parallel machines by minimizing the total weighted completion time. *Comput. Oper. Res.* **123** (2020) 105018.

[29] A. Kramer, M. Iori and P. Lacomme, Mathematical formulations for scheduling jobs on identical parallel machines with family setup times and total weighted completion time minimization. *Eur. J. Oper. Res.* **289** (2021) 825–840.

[30] R. Kramer and A. Kramer, An exact framework for the discrete parallel machine scheduling location problem. *Comput. Oper. Res.* **132** (2021) 105318.

[31] F.K. Lemos, A.C. Cherri and S.A. de Araujo, The cutting stock problem with multiple manufacturing modes applied to a construction industry. *Int. J. Prod. Res.* **59** (2021) 1088–1106.

[32] F.K. Lemos, A.C. Cherri, S.A. de Araujo and H.H. Yanasse, Minimizing saw cycles on the cutting stock problem with processing times depending on the cutting pattern. *J. Oper. Res. Soc.*, accepted (2022).

[33] N. Ma, Y. Liu and Z. Zhou, Two heuristics for the capacitated multi-period cutting stock problem with pattern setup cost. *Comput. Oper. Res.* **109** (2019) 218–229.

[34] N. Maculan, M.d.M. Passini, J.A.d.M. Brito and I. Loiseau, Column-generation in integer linear programming. *RAIRO: Oper. Res.* **37** (2003) 67–83.

[35] J. Martinovic, G. Scheithauer and J.M. Valério de Carvalho, A comparative study of the arcflow model and the one-cut model for one-dimensional cutting stock problems. *Eur. J. Oper. Res.* **266** (2018) 458–471.

[36] J. Martinovic, M. Delorme, M. Iori, G. Scheithauer and N. Strasdat, Improved flow-based formulations for the skiving stock problem. *Comput. Oper. Res.* **113** (2020) 104770.

[37] G.M. Melega, S.A. de Araujo and R. Jans, Classification and literature review of integrated lot-sizing and cutting stock problems. *Eur. J. Oper. Res.* **271** (2018) 1–19.

[38] G.M. Melega, S.A. de Araujo and R. Morabito, Mathematical model and solution approaches for integrated lot-sizing, scheduling and cutting stock problems. *Ann. Oper. Res.* **295** (2020) 695–736.
[39] F. Parreño, M.T. Alonso and R. Alvarez-Valdés, Solving a large cutting problem in the glass manufacturing industry. *Eur. J. Oper. Res.* **287** (2020) 378–388.
[40] K.C. Poldi and S.A. de Araujo, Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem. *Ann. Oper. Res.* **238** (2016) 497–520.
[41] S.C. Poltroniere, K.C. Poldi, F.M.B. Toledo and M.N. Arenales, A coupling cutting stock-lot sizing problem in the paper industry. *Annals of Oper. Res.* **157** (2008) 91–104.
[42] B. Ramos, C. Alves and J.M. Valério de Carvalho, An arc flow formulation to the multitrip production, inventory, distribution, and routing problem with time windows. *Int. Trans. Oper. Res.* **29** (2022) 526–553. DOI: 10.1111/itor.12765.
[43] M. Saeedi and R. Feizi, Modeling and optimization of batch production based on layout and cutting problems under uncertainty. *RAIRO: Oper. Res.* **55** (2021) 61–81.
[44] M.C. Santoro and F.K. Lemos, Irregular packing: Milp model based on a polygonal enclosure. *Ann. Oper. Res.* **235** (2015) 693–707.
[45] C.d.A. Signorini, S.A. de Araujo and G.M. Melega, One-dimensional multi-period cutting stock problems in the concrete industry. *Int. J. Prod. Res.* **60** (2022) 2386–2403.
[46] Y. Song, J. Zhang, Z. Liang and C. Ye. An exact algorithm for the container drayage problem under a separation mode. *Transp. Res. Part E: Log. Transp. Rev.* **106** (2017) 231–254.
[47] R.S. Trindade, O.C.B. de Araújo and M. Fampa, Arc-flow approach for single batch-processing machine scheduling. *Comput. Oper. Res.* **134** (2021) 105394.
[48] J.M. Valério de Carvalho, Exact solution of bin-packing problems using column generation and branch-and-bound. *Ann. Oper. Res.* **86** (1999) 629–659.
[49] J.M. Valério de Carvalho, LP models for bin packing and cutting stock problems. *Eur. J. Oper. Res.* **141** (2002) 253–273.
[50] P.H. Vance, Branch-and-price algorithms for the one-dimensional cutting stock problem. *Comput. Optim. Appl.* **9** (1998) 211–228.
[51] G. Wäscher, H. Haußner and H. Schumann, An improved typology of cutting and packing problems. *Eur. J. Oper. Res.* **183** (2007) 1109–1130.
[52] L.A. Wolsey, Valid inequalities, covering problems and discrete dynamic programs, in *Annals of Discrete Mathematics*. Vol. 1. Elsevier (1977) 527–538.
[53] D.A. Wuttke and H. S. Heese, Two-dimensional cutting stock problem with sequence dependent setup times. *Eur. J. Oper. Res.* **265** (2018) 303–315.
[54] H.H. Yanasse and M.J.P. Lamosa, An integrated cutting stock and sequencing problem. *Eur. J. Oper. Res.* **183** (2007) 1353–1370.