Mathematical models for the minimization of open stacks problem

Mateus Martin[a,*], Horacio Hideki Yanasse[a], Maria José Pinto[b]

[a] *Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, Brazil*
[b] *Divisão C4ISR, Instituto de Estudos Avançados (IEAv), Brazil*

Abstract

In this paper, we address the Minimization of Open Stacks Problem (MOSP). This problem often appears during production planning of manufacturing industries, such as in the cutting of objects to comply with space constraints around the cutting machine in the glass, furniture, and metallurgical industries. The MOSP is also pertinent to the field of VLSI design and Graph theory. The MOSP consists of finding an optimal sequence of a given set of patterns, while minimizing the maximum number of simultaneously open stacks. To effectively model and solve the problem, we present a novel Integer Linear Programming (ILP) formulation based on a graph representation of the problem. We derive an ILP formulation from the modeling approach of Faggioli & Bentivoglio (1998) for the MOSP. Then we develop a simple Constraint Programming (CP) model based on interval variables and renewable resources. We performed computational experiments to evaluate the proposed approaches in comparison with other ILP formulations from the literature. The proposed approaches are competitive in solution quality and computational time for small and moderate sized problem instances.

Keywords: Pattern sequencing problems, Integer linear programming, MOSP graph, Constraint programming, Interval variables

1. Introduction

The manufacturing industries usually have a stage of production with several cutting operations decisions to be made. These operations can be seen, for example, in the cutting of glass (Dyson & Gregory, 1974; Madsen, 1979), steel bars (Armbruster, 2002), and paper (Matsumoto et al., 2011). In Operational Research, the cutting operations are addressed by optimization problems related to pattern generation, pattern minimization, and pattern sequencing. The pattern generation problems seek to establish how a set of objects should be geometrically cut to produce a set of items to minimize waste and satisfy the constraints of the cutting machine – see Wäscher et al. (2007) for a complete survey and typology of the so-called Cutting & Packing (C&P) problems. Next, the pattern minimization problems seek to minimize the number of different patterns to reduce the idle time of the cutting machine due to setups, where the prioritizing of time may lead to increased waste as the economic tradeoff (Cui et al., 2015; Martin et al., 2018). Finally,

*Corresponding author. Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, Brazil, Avenida Cesare Mansueto Giulio Lattes, 1201, 12247-014, Eugênio de Melo, São José dos Campos, SP, Brazil.

Email addresses: mpmartin@unifesp.br (Mateus Martin), horacio.yanasse@unifesp.br (Horacio Hideki Yanasse), maju@ieav.cta.br (Maria José Pinto)

the pattern sequencing problems seek to determine the processing sequence of the patterns, while minimizing an objective function, e.g., the maximum number of simultaneously open stacks, or the total sum of the lifespan of the stacks. These three classes of problems are usually solved sequentially, despite some integrated approaches in the literature (Armbruster, 2002; Yanasse & Lamosa, 2007; Rinaldi & Franz, 2007).

Thus, the Minimization of Open Stack Problem (MOSP) is a pattern sequencing problem that addresses in-process inventory decisions. During the processing of the patterns, all the copies of an item are placed close to the cutting machine in a specific space known as a stack. The stack of an item is open if the production of this item has been started, but has not yet been finished. After the processing of all the patterns that produce this item, the corresponding stack is closed and dispatched to the next stage of production. The MOSP consists of determining a processing sequence of the patterns that minimizes the maximum number of simultaneously open stacks. This is important when there are space constraints around the cutting machine or its number of automatic compartments is limited (Yuen, 1991; Armbruster, 2002). Notice that one could move incomplete stacks to other areas of the plant when there is no more space near the cutting machine. However this is avoided due to handling costs (Yanasse & Senne, 2010). Examples of other pattern sequencing problems include the Minimization of Order Spread Problem (MORP) which minimizes the total sum of the lifespan of the stacks (Foerster & Wäscher, 1998; De Giovanni et al., 2013b), the Minimization of Tool Switches Problems (MTSP) which minimizes the number of tool switches in a machine of a flexible manufacturing system (Tang & Denardo, 1988; Silva et al., 2020), and the Minimization of Discontinuities Problem which is also known as the Consecutive Blocks Minimization Problem (Soares et al., 2020).

1.1. Related work

The list of approaches for the MOSP is not extensive; however the problem has been addressed with several different methods in the last decades. Yuen (1991, 1995) seem to have published the first approaches in the literature to address the MOSP, motivated by an application in an Australian glass industry. The author proposed six constructive heuristics for the problem, based on simple criteria to sequence the next pattern from the current set of open stacks. Then, Yuen & Richardson (1995) presented a trivial lower bound for the MOSP, given by the maximum number of different items in the patterns. They also developed an exact method that enumerates the permutations of patterns.

Relevant theoretical results for the MOSP involve modeling using graphs (Yanasse, 1997a; Lopes & De Carvalho, 2015), the proof of its NP-Hardness (Yanasse, 1997b), several lower bounds with emphasis on the arc contraction heuristic (Yanasse et al., 1999), the identification of some special structures of the problem exactly solved by polynomial time algorithms (Yanasse & Limeira, 2004), the development of Dynamic Programming formulations (Deo et al., 1987; De La Banda & Stuckey, 2007; Chu & Stuckey, 2009), and the use of preprocessing operations seeking to downsize the problem instances (Yanasse & Senne, 2010). Yanasse (1997a) proved that any problem instance of the MOSP can be modelled as the MOSP graph, in which the nodes represent the items, and there is an arc connecting two nodes, if, and only if, there is at least one pattern producing the both corresponding items. Thus, the problem can be seen as a Graph Traversing problem. Becceneri et al. (2004) developed the Minimal Cost Node heuristic, using the idea of traversing the arcs in the MOSP graph by considering the least quantity of arcs to be sequenced as criterion in order to close the node. A node is “closed” when all the arcs incident to it are traversed; and a node is “open” after the first arc incident to it is traversed. Thus, it is solution-wise equivalent for the MOSP to sequence the patterns or to sequence the closed stacks (i.e., the nodes of the MOSP graph).

Several heuristics and meta-heuristics have been proposed for the MOSP in the literature. As the problem has several alternative solutions, and any sequence of the patterns (or of the closed stacks) is a feasible solution to the problem, these heuristic approaches often find optimal or near-optimal solutions in short running times. There are solution approaches based on Tabu search (Faggioli & Bentivoglio, 1998; Fink & Voß, 1999), microcanonical optimization (Linhares & Yanasse, 2002), genetic algorithms (De Oliveira & Lorena, 2002; De Giovanni et al., 2013a), constructive heuristics based on Hamiltonian circuits (Ashikaga & Soma, 2009), breadth-first search (Carvalho & Soma, 2015), biased random-key genetic algorithm (Gonçalves et al., 2016), variable neighborhood descent search (Lima & Carvalho, 2017), adaptive large neighborhood search (Santos & Carvalho, 2018), and page rank (Frinhani et al., 2018).

We are aware of three Integer Linear Programming (ILP) formulations for the MOSP. First Yanasse (1997b) proposed an ILP formulation from a previous model proposed in Tang & Denardo (1988) for the MTSP, after proving the relations of both problems. The MOSP was the theme problem in the *First Constraint Modeling Challenge* (Smith & Gent, 2005), where the entry of Baptiste (2005) proposed an ILP position-sequence formulation to sequence the patterns. More recently, Lopes & De Carvalho (2015) proposed an interesting ILP formulation based on interval graphs and completing edges, in which an interval of time is linked to each open stack. To the best of our knowledge, there is no approach in the literature comparing the relative computational performance of these formulations. However, the analysis of these formulations and their constraints is relevant, since limiting the maximum number of simultaneously open stacks is also considered as a constraint in some approaches (Belov & Scheithauer, 2007; Matsumoto et al., 2011; De Giovanni et al., 2013b).

1.2. Our contributions

The main contributions that are presented in this paper are: (a) the proposition of a novel ILP formulation for the MOSP sequencing the closed stacks, inspired by the MOSP graph; (b) the proposition of an ILP formulation for the MOSP sequencing the patterns, derived from the approach of Faggioli & Bentivoglio (1998) for enumerating the number of simultaneously open stacks; and, (c) the presentation of a Constraint Programming (CP) model for the MOSP sequencing of the patterns, based on interval variables. In advance, we highlight that our mathematical models perform best for scenarios characterized by a small/moderate number of patterns or for scenarios that lead to a dense MOSP graph, but are not competitive with the best algorithms of the literature (Chu & Stuckey, 2009; Gonçalves et al., 2016). We highlight the importance of developing new models for pattern sequencing problems, since they may contribute to the understanding of the problems, and inspire strategy solutions in the context of general-purpose solvers. Furthermore, these models can be integrated into other optimization problems to solve even complex problems, e.g., considering also decisions of pattern generation and pattern minimization problems.

1.3. Organization of the paper

The remainder of this paper is organized as follows. In Section 2, we formally state the MOSP and present an illustrative example. In Section 3, we propose two ILP formulations for the MOSP, one sequencing the closed stacks and the other sequencing the patterns. In Section 4, we develop a CP model for the MOSP to sequence the patterns based on interval variables. The computational results presented in Section 5 show the main advantages/difficulties of each approach. In Section 6, we present final remarks and discuss future research.

2. Description of the problem

We formally describe the MOSP by a binary matrix A of dimension $M \times N$. For cutting operations, each entry a_{ij} of matrix A equals 1 if item (=row) i is produced by pattern (=column) j, and 0 otherwise. The number of copies of item i produced by pattern j is not relevant. Any feasible solution of the MOSP corresponds to a permutation $\Pi = (\pi_1, \ldots, \pi_N)$ of the columns of matrix A, where π_j, $1 \leq j \leq N$, represent the column positioned in the jth position; that is, the new jth column of the matrix is $(a_{1,\pi_j} \ldots a_{M,\pi_j})^T$. Thus, there are $N!$ permutations in the solution space. The MOSP consists of finding a processing sequence of the patterns (i.e., a permutation Π) that minimizes the maximum number of simultaneously open stacks. More formally, the stack of item i (or stack i) is open during the kth position/instant of the sequence of patterns, if $\sum_{j=1}^{k} a_{i,\pi_j} \cdot \sum_{j=k}^{N} a_{i,\pi_j} > 0$ holds. Thus, the maximum number of simultaneously open stacks of a sequence Π is given by

$$MOSP(\Pi) = \max_{k:1 \leq k \leq N} \Big\{ i : \sum_{j=1}^{k} a_{i,\pi_j} \cdot \sum_{j=k}^{N} a_{i,\pi_j} > 0 \Big\}.$$

The MOSP has a wide range of real-life applications apart from cutting operations, since the rows and columns of matrix A can represent, respectively, order types and groupings of these order types. See Linhares & Yanasse (2002) for a list of 12 related problems to the MOSP.) For instance, in the field of Very Large Scale Integration (VLSI) design, the MOSP is known as the Gate Matrix Layout Problem, where the patterns represent the vertical gates, the items represent the nets, and the stacks represent the horizontal tracks, thereby reducing the maximum number of simultaneously open stacks, which helps to downsize the area of the circuits (Linhares et al., 1999).

In Table 1, we show an illustrative problem instance of the MOSP with $M = 5$ items and $N = 7$ patterns. This instance was proposed in Costa et al. (2017), who developed MOSP-based approaches to optimize in-process inventory for the main automotive manufacturer of the Brazilian market. For this industrial application, the patterns represent vehicles, and the items represent optional products of the vehicles; thus, the stacks are formed at the beginning of the stage of production, instead of at the end of the stage of production, as occurs in a cutting operations setting. Therefore, items I1 and I3 are embedded in vehicle P5, or from a cutting operations perspective, pattern P5 produces items I1 and I3.

In Table 2, we show two sequences of patterns for the problem instance presented in Table 1. Symbol * in entry a_{i,π_j} points out that pattern π_j does not produce item i, although the production of item i has been started but not yet finished during the processing of this pattern. This recalls the Consecutive One property of the MOSP (Yanasse, 1997a). Sequence $\Pi_A = \{$P2,P3,P5,P1,P4,P6,P7$\}$ has value of 4 for the MOSP; sequence $\Pi_B = \{$P5,P1,P2,P6,P3,P7,P4$\}$ is optimal with value of 3, which happens during instants 2, 4 and 6. For instance, the number of open stacks during the processing of pattern P5 in sequence Π_A is 4 (stacks I1, I2, I3, and I4 are open), while in sequence Π_B is 2 (stacks I1 and I3 are open).

It is a solution-wise equivalent for the MOSP to sequence the patterns or to sequence the closed stacks. We next make use of the ideas of the MOSP graph to sequence the closed stacks, instead of the patterns. Note, in Table 2, that when processing the sequence of patterns Π_B, the stacks of

Table 1: Illustrative problem instance of the MOSP.

	Patterns						
Items	P1	P2	P3	P4	P5	P6	P7
I1	1	0	0	0	1	0	0
I2	1	1	0	0	0	1	0
I3	0	1	1	0	1	0	1
I4	0	0	1	1	0	1	0
I5	0	0	0	1	0	0	1

Table 2: Two sequences of patterns for the problem instance presented in Table 1.

	Sequence Π_A							Sequence Π_B						
Items	P2	P3	P5	P1	P4	P6	P7	P5	P1	P2	P6	P3	P7	P4
I1	0	0	1	1	0	0	0	1	1	0	0	0	0	0
I2	1	*	*	1	*	1	0	0	1	1	1	0	0	0
I3	1	1	1	*	*	*	1	1	*	1	*	1	1	0
I4	0	1	*	*	1	1	0	0	0	0	1	1	*	1
I5	0	0	0	0	1	*	1	0	0	0	0	0	1	1

$a_{i,\pi_j} = *$ recalls the Consecutive One property of the MOSP.

the items are closed in the following order: I1 (after the processing of P1), I2 (after the processing of P6), I3 (after the processing of P7), and I4 and I5 (after the processing of P4). Recall that the MOSP graph is a graph in which the nodes represent the items, and there is an arc connecting two nodes, if, and only if, there is at least one pattern producing both corresponding items. Thus, the MOSP graph is a union of cliques, given that a pattern that produces k items is represented by a clique of size k. In Fig. 1, we illustrate the MOSP graph of the problem instance presented in Table 1.

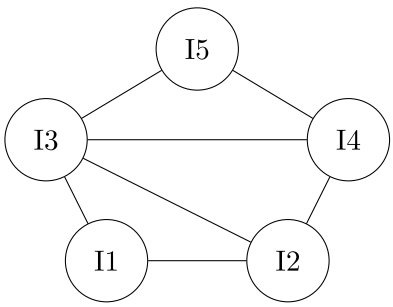

Figure 1: Illustration of the MOSP graph of the problem instance presented in Table 1.

We define the following sets to be used when presenting the models:

$I = \{1, \ldots, M\}$	set of items/stacks;
$J = \{1, \ldots, N\}$	set of patterns;
$J_i = \{j \in J \mid a_{ij} = 1\}$	set of all patterns that produce item i, $i \in I$.

Note that all patterns $j \in J_i$ must be processed to close the stack of item $i \in I$. Consequently, the stacks of the other items produced by these $|J_i|$ patterns have all been opened and may even have been closed, depending on the sequence of patterns. As a general rule, other patterns $j \notin J_i$ can be sequenced within the sub-sequence of patterns $j \in J_i$ needed to close the stack of item $i \in I$. However, when introducing the MOSP as a problem of traversing arcs in a graph, Yanasse (1997a) proved that there always exists an optimal sequence of patterns in which there is no pattern $j \notin J_i$ within the sub-sequence of patterns of the current stack to be closed. Thus, without loss of optimality, we can narrow the search to this specific type of sequences of patterns. Moreover, we can always generate a sequence of patterns from a sequence of closed stacks (and vice-versa), with both sequences providing the same maximum number of open stacks.

Therefore, in a sequence of closed stacks, we consider that when a node/stack is closed, then all its adjacent nodes/stacks in the MOSP graph have already been opened. For instance, the analysis of Fig. 1 shows that when the stack of item I3 is closed, we know that all the other four stacks have already been opened, because of the adjacency of node/item I3 with all other nodes/items in the MOSP graph. Thus, the sequences of closed stacks $\Phi_{B1} = \{\text{I1,I2,I3,I5,I4}\}$ and $\Phi_{B2} = \{\text{I1,I2,I3,I4,I5}\}$ are equivalent to the sequence of patterns $\Pi_B = \{\text{P5,P1,P2,P6,P3,P7,P4}\}$, with a value of maximum number of simultaneously open stacks equals to 3 stacks.

3. Two ILP formulations for the MOSP

In this section, we present two ILP formulations for the MOSP. The formulation in Section 3.1 sequences the closed stacks, and formulation in Section 3.2 sequences the patterns.

3.1. An ILP formulation sequencing the closed stacks

We denote by in-process stacks the open or closed stacks, until the end of the sequence of closed stacks. Therefore, the number of open stacks at any position/instant of the sequence of closed stacks is equal to the current number of in-process stacks minus the current number of closed stacks. Note that the number of closed stacks always increases by one unit at each instant in a sequence of closed stacks.

We next formulate the MOSP as an ILP formulation, which is denoted by MOSP-ILP-I. The formulation has three sets of variables. The first one concerns the sequence of closed stacks, the second one refers to the in-process stacks, and the last one counts the maximum number of simultaneously open stacks. Thus, we define the following sets and variables as:

$T = \{1, \ldots, M\}$	set of instants to close the stacks;
S_i	closed neighborhood of node $i \in I$ in the MOSP graph, that is, the set of all adjacent nodes of node $i \in I$ in which node i itself is included;
x_{it}	binary variable which equals 1, if stack $i \in I$ is closed at instant $t \in T$, and 0 otherwise;
w_{it}	binary variable which equals 1. if stack $i \in I$ is in-process (i.e., open or closed) at instant $t \in T \backslash \{M\}$, and 0 otherwise;
C	integer variable that counts the maximum number of simultaneously open stacks.

We assume, without loss of optimality, that all stacks are in-process during instant $t = M$. Formulation MOSP-ILP-I is given by model (1):

$$\textbf{Min} \quad C. \tag{1a}$$

subject to

$$\sum_{i \in I} x_{it} = 1, \qquad t \in T. \quad (1b)$$

$$\sum_{t \in T} x_{it} = 1, \qquad i \in I. \quad (1c)$$

$$\sum_{k \in S_i} \sum_{t'=1}^{t} x_{kt'} \leq \min\{t, |S_i|\} w_{it}, \qquad i \in I, t \in T \backslash \{M\}. \quad (1d)$$

$$C \geq \sum_{i \in I} w_{it} - t + 1, \qquad t \in T. \quad (1e)$$

$$x_{it} \in \{0, 1\}, \qquad i \in I, t \in T. \quad (1f)$$

$$w_{it} \in \{0, 1\}, \qquad i \in I, t \in T \backslash \{M\}. \quad (1g)$$

$$C \in \mathbb{Z}_+. \quad (1h)$$

$$w_{iM} = 1, \qquad i \in I. \quad (1i)$$

The objective function (1a) minimizes the maximum number of simultaneously open stacks. Constraints (1b) and (1c) ensure a sequence for closing the stacks. Constraints (1d) are disjunctive inequalities of the Big-M type, in which $\min\{t, |S_i|\}$ denotes a sufficiently large number. These constraints ensure that, if a stack $k \in S_i$ is closed at an instant $t' \in T \backslash \{M\}$, $1 \leq t' \leq t$, then stack $i \in I$ is in-process at instant t. Additionally, constraints (1d) ensure that, when a stack $i \in I$ is closed at an instant, then all its adjacent nodes/stacks (i.e., in S_i) in the MOSP graph have been previously opened. Constraints (1e) count the maximum number of simultaneously open stacks. For each instant $t \in T$, the number of open stacks is given by the number of in-process stacks during instant t (i.e., $\sum_{i \in I} w_{it}$) minus the number of closed stacks at instant t (i.e., $t - 1$). Constraints (1f)–(1h) define the domain of the variables. Constraints (1i) work as fixing variables, since all stacks are in-process at instant $t = M$. The number of variables in this model is $2M^2 - M + 1$, while the number of constraints is $M^2 + 2M$. We note that a previous version of model (1) was introduced in a conference, albeit with limited computational experiments (Yanasse & Pinto, 2003) (in Portuguese), in which the value of Big-M in constraints (1d) was M (instead of $\min\{t, |S_i|\}$). The value in our model substantially improves the linear relaxation of the model.

Recall that the sequence of closed stacks Φ_{B1} provides the optimal value of up to 3 simultaneously open stacks for the problem instance of the MOSP presented in Table 1. In Fig. 2, using the MOSP graph, we illustrate the optimal sequence $\Phi_{B1} = \{\text{I1,I2,I3,I5,I4}\}$. Although sets I and J are numbered from 1 to M and N, respectively, in what follows we chose to indicate the items (resp. patterns) with a prefix “I” (resp. “P”), as appears in Table 2, to clarify the explanation. In Formulation MOSP-ILP-I, we have $x_{\text{I1},1} = x_{\text{I2},2} = x_{\text{I3},3} = x_{\text{I5},4} = x_{\text{I4},5} = 1$ for this sequence; the value of all other variables x_{it} is zero. In particular, $w_{\text{I1},t} = w_{\text{I2},t} = w_{\text{I3},t} = 1$ with $t = 1, \ldots, M$, because $x_{\text{I1},1} = 1$, and nodes I2 and I3 are adjacent to node I1 in the MOSP graph. During instant $t = 2$, the number of in-process stacks is 4 (i.e., $\sum_{i \in I} w_{i2} = 4$ given by stacks I1, I2, I3, and I4), and the number of closed stacks is 1 (i.e., stack I1); thus, the number of simultaneously open stacks is 3 (i.e., stacks I2, I3, and I4). In contrast, during instant $t = 5$, the number of in-process stacks is 5 (i.e., $\sum_{i \in I} w_{i5} = 5$ given by all the five stacks), and the number of closed stacks is 4 (i.e., given by stacks I1, I2, I3, and I5); thus, the number of simultaneously open stacks is 1 (i.e., stack I4).

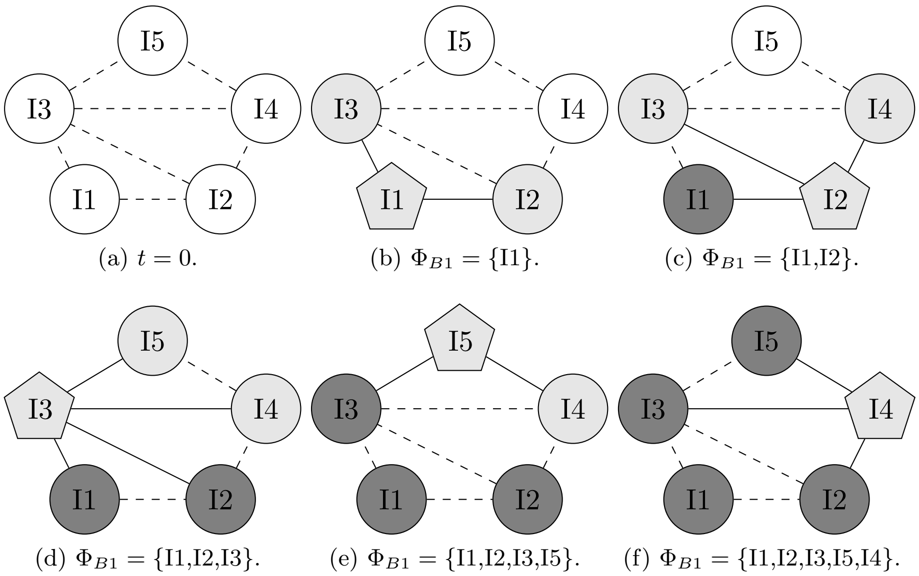

Figure 2: Illustration of optimal sequence $\Phi_{B1} = \{I1,I2,I3,I5,I4\}$ in the MOSP graph. The pentagon nodes are the stacks to be closed in the current instant. The light grey nodes are the simultaneously open stacks during the current instant, while the dark grey nodes are the previously closed stacks.

3.1.1. Enhancing Formulation MOSP-ILP-I

In what follows, we propose the use of a lower bound for the MOSP to reduce the number of variables/constraints in Formulation MOSP-ILP-I by downsizing the set of instants to close the stacks. Reducing the number of variables/constraints in an ILP formulation is usually helpful to accelerate the convergence of the algorithms in the context of general-purpose ILP solvers. So, we define the lower bound for the MOSP as lb, where $C \geq lb$.

Recall that there are $t-1$ closed stacks at an instant $t \in \{1,\ldots,M\}$ in a sequence of closed stacks; thus, there are up to $M-(t-1)$ simultaneously open stacks at this instant. Specifically, we have just one open stack at instant $t = M$, up to two open stacks at instant $t = M-1$, up to three open stacks at instant $t = M-2$, and so on. Please see Fig. 2.) As a general rule, we have up to lb open stacks at instant $t = M-(lb-1)$. Since $C \geq lb$, the maximum number of simultaneously open stacks necessarily happens prior to, or at, instant $t = M-(lb-1)$. Thus, without loss of optimality, we can disregard instants $t \in T \mid t > M-(lb-1)$. Previously, in Formulation MOSP-ILP-I, we considered that the set of instants was $T = \{1,\ldots,M\}$, or equivalent to the set of items/stacks. We now redefine set T to be:

$T = \{1,\ldots,\overline{T}\}$ the improved set of instances to close the stacks, where $\overline{T} = M-(lb-1)$.

Additionally, we assume that one stack is closed at any instant $t \in T \backslash \{\overline{T}\}$, and that lb stacks are closed at instant $t = \overline{T}$, in a total of M stacks. Formulation MOSP-ILP-I can be enhanced by replacing constraints (1b) to constraints (2). The other variables/constraints of Formulation MOSP-ILP-I are kept as previously defined, within the redefinition of set T.

$$\sum_{i \in I} x_{it} = 1, \qquad t \in T \backslash \{\overline{T}\}. \quad (2a)$$

$$\sum_{i \in I} x_{i\overline{T}} = lb. \tag{2b}$$

Constraints (2a) ensure that only one stack is closed at any instant $t \in T \backslash \{\overline{T}\}$. Constraint (2b) ensures that lb stacks are closed at instant $t = \overline{T}$. In addition to the maximum number of different items in the patterns, another trivial lower bound for the MOSP is given by the smallest degree of a node in the MOSP graph plus one unit (Yanasse et al., 1999). This lower bound provides the value of 3 for the MOSP graph in Fig. 1, for instance, which matches the value of the optimal solution for the corresponding problem instance. For our computational experiments, we applied constraints (2) in Formulation MOSP-ILP-I with the lower bound of the smallest degree of a node in the MOSP graph plus one unit.

3.2. An ILP formulation sequencing the patterns

We next derive an ILP formulation for the MOSP, which is denoted by MOSP-ILP-II, based on the modeling approach of Faggioli & Bentivoglio (1998). These authors proposed a way to count the number of simultaneously open stacks for the MOSP from a fixed sequence of patterns. They noted that, after processing each pattern, the corresponding column of matrix A can be added to a vector of integer variables. As a general result, at the end of the sequence of patterns, we find that each item $i \in I$ was produced by $|J_i|$ patterns. Note that the set of instants to process a pattern is equal to the set of patterns J.

The formulation has five sets of variables. The first one concerns the sequence of patterns, and the second one corresponds to the vectors of integer variables. The third and fourth ones link these vectors of integer variables with the in-process and closed stacks, respectively. The last one counts the maximum number of simultaneously open stacks. Thus we define the following variables:

- q_{jt} binary variable which equals 1, if pattern $j \in J$ is processed at instant $t \in J$, and 0 otherwise;
- f_{it} integer variable that counts the number of patterns $j \in J_i$, $i \in I$, already sequenced up to instant $t \in J$;
- g_{it} binary variable that equals 1, if stack $i \in I$ is in-process (i.e., open or closed) at instant $t \in J$, and 0 otherwise;
- h_{it} binary variable that equals 1, if stack $i \in I$ is not closed at instant $t-1$, $t \in J$, and 0 otherwise.

Formulation MOSP-ILP-II is given by model (3). We note that variables q_{jt}, g_{it}, and h_{it}, and the following constraints, (3a)–(3b) and (3d)–(3e), correspond to our adjustments to obtain a linear model.

Min (1a).

subject to

(1h).

$$\sum_{t \in J} q_{jt} = 1, \qquad j \in J. \tag{3a}$$

$$\sum_{j \in J} q_{jt} = 1, \qquad t \in J. \tag{3b}$$

$$f_{it} = f_{i(t-1)} + \sum_{j \in J_i} q_{jt}, \qquad i \in I, t \in J. \tag{3c}$$

$$f_{it} \leq \min\{|J_i|, t\} g_{it}, \qquad i \in I, t \in J. \quad (3d)$$
$$|J_i| - f_{i(t-1)} \leq \min\{|J_i|, N - t + 1\} h_{it}, \qquad i \in I, t \in J. \quad (3e)$$
$$C \geq \sum_{i \in I} g_{it} + \sum_{i \in I} h_{it} - M, \qquad t \in J. \quad (3f)$$
$$\sum_{t=1}^{\lfloor N/2 \rfloor} q_{kt} = 1. \quad (3g)$$
$$q_{jt} \in \{0, 1\}, \qquad j \in J, t \in J. \quad (3h)$$
$$f_{it} \in \mathbb{Z}_+, g_{it} \in \{0, 1\}, \qquad i \in I, t \in J \backslash N. \quad (3i)$$
$$h_{it} \in \{0, 1\}, \qquad i \in I, t \in J. \quad (3j)$$
$$f_{i0} = 0, f_{iN} = |J_i|, g_{iN} = 1, \qquad i \in I. \quad (3k)$$

Constraints (3a) and (3b) ensure a processing sequence of the patterns. Constraints (3c) ensure the update of the number of sequenced patterns that produce item $i \in I$ up to instant $t \in J$. Constraints (3d) and (3e) are disjunctive inequalities of the Big-M type with the sufficiently large numbers already indicated. Constraints (3d) enforce variables g_{it} to assume the value of 1, when $f_{it} > 0$. Note that the number of in-process stacks after the processing of the pattern sequenced at instant $t \in J$ is given by $\sum_{i \in I} g_{it}$. Constraints (3e) allow variables h_{it} to assume the value of 0 only when $f_{i(t-1)} = |J_i|$. Note that the number of closed stacks before the processing of the pattern sequenced at instant $t \in J$ is given by $M - \sum_{i \in I} h_{it}$. Constraint (3f) count the number of simultaneously open stacks. For an instant $t \in J$, the number of simultaneously open stacks is given by the number of in-process stacks (i.e., $\sum_{i \in I} g_{it}$) minus the number of closed stacks (i.e., $M - \sum_{i \in I} h_{it}$). Valid inequality (3g) ensures that pattern k is sequenced in one of the first $\lfloor N/2 \rfloor$ positions/instants of the sequence of patterns. Constraints (3h)–(3j) define the domain of the variables. Constraints (3k) work as fixing variables. The objective function and integer variable C are kept as defined in Formulation MOSP-ILP-I. The number of variables in this model is $3MN + N^2 - 2M + 1$, while the number of constraints is $3MN + 3N$. In Table 3, we illustrate the values of variables f_{it} for the sequence of patterns Π_B.

Table 3: Illustration of the values of variables f_{it} for the sequence of patterns Π_B.

			Sequence Π_B						
Items	$\lvert J_i \rvert$	$t = 0$	P5	P1	P2	P6	P3	P7	P4
I1	2	**0**	1	2	2	2	2	2	**2**
I2	3	**0**	0	1	2	3	3	3	**3**
I3	4	**0**	1	1	2	2	3	4	**4**
I4	3	**0**	0	0	0	1	2	2	**3**
I5	2	**0**	0	0	0	0	0	1	**2**

Note that, in Table 3, the entries of $f_{i0} = 0$ and $f_{iN} = |J_i|$, $i \in I$, are highlighted in bold. In Formulation MOSP-ILP-II, we have $q_{\text{P5},1} = q_{\text{P1},2} = q_{\text{P2},3} = q_{\text{P6},4} = q_{\text{P3},5} = q_{\text{P7},6} = q_{\text{P4},7} = 1$; the value of all other variables q_{jt} is 0. In particular, during instant $t = 2$ (processing of P1), the number of in-process stacks is 3 (i.e., $\sum_{i \in I} g_{i2} = 3$ given by stacks I1, I2 and I3), and the number of closed stacks is 0 (i.e., $M - \sum_{i \in I} h_{i2} = 0$); thus, the number of simultaneously open stacks is 3. In contrast, during instant $t = 7$ (processing of P4), the number of in-process stacks is 5 (i.e., $\sum_{i\in I} g_{i7} = 5$ given by all the five stacks), and the number of closed stacks is 3 (i.e., $M - \sum_{i\in I} h_{i7} = 3$ given by stacks I1, I2 and I3); thus, the number of simultaneously open stacks is 2 (i.e., stacks I3 and I4, which are then closed after $t = 7$).

4. A CP model for the MOSP

In this section, we develop a simple CP model for the MOSP sequencing the patterns, denoted by MOSP-CP. We interpret the MOSP as a scheduling problem involving interval variables and renewable resources (cumulative resources). In essence, an interval variable expresses an interval of time during which a task is performed, whose position in time is an unknown of the problem. An interval variable is characterized by a start value, an end value, and a size. The space around the cutting machine is modelled as a renewable resource, that is, during the production of item $i \in I$, one stack is formed, consuming one unit of the resource. Thus, when the production of this item is finished, the resource is freed to be used again, if necessary; thereby, physically representing the use of space around the cutting machine.

Model MOSP-CP has two sets of interval variables. The first one concerns the set of patterns. The second one refers to the set of items, wherein each item will form a stack. Thus, we define the following interval variables as:

$pattern_j$ interval variable of pattern $j \in J$ with a possible start value of 0, a possible end value of N, and a fixed size of 1 unit (i.e., the pattern is processed during one instant); and

$item_i$ interval variable of item/stack $i \in I$ with a possible start value of 0, a possible end value of N, and a size varying from $|J_i|$ to N units.

Using the OPL modeling language (Van Hentenryck, 1999), we formulate Model MOSP-CP as given by model (4).

Min (1a).

subject to

(1h).

$$\texttt{noOverlap}(pattern_1, \ldots, pattern_N). \tag{4a}$$

$$\texttt{span}(item_i, \{pattern_j, j \in J_i\}), \qquad i \in I. \tag{4b}$$

$$C \geq \sum_{i\in I} \texttt{pulse}(item_i, 1). \tag{4c}$$

Constraints (4a) ensure that the patterns will be sequenced in some order, as predicate `noOverlap` provides the non-overlapping of a set of interval variables. Constraints (4b) link the interval of time of the stack of item $i \in I$ with the intervals of time of all patterns that produce this item (i.e., $\forall j \in J_i$). In fact, predicate `span` enforces that $item_i$ starts with the first pattern $j' \in J_i$ in the sequence of patterns, and ends with the last pattern $j'' \in J_i$ in the sequence of patterns. Constraint (4c) enforces that the maximum number of simultaneously open stacks is equal or greater than the maximum amount of resource used. The number of variables in this model is $N + M + 1$, while the number of constraints is $M + 2$.

In Fig. 3, we illustrate the predicates `noOverlap`, `span`, and `pulse`, using the sequence of patterns Π_B. Predicate `noOverlap` generates the sequence of patterns, since each pattern $j \in J$

corresponds to an interval variable with unitary length, and these intervals should not overlap in any of the N instants. Predicate `span` enforces the link of the intervals among an item $i \in I$ and its patterns $j \in J_i$. Note that patterns P1 and P6 are sequenced during the processing of item I3; however, they do not produce copies of this item. Predicate `pulse` provides a profile of the maximum number of units of consumed resource; in other words, the simultaneously open stacks in each instant of the sequence of patterns.

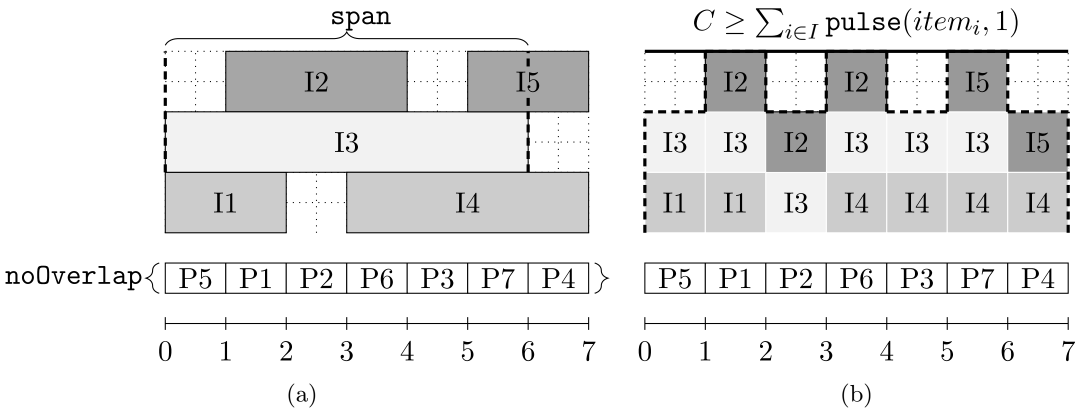

Figure 3: Illustration of predicates (a) `noOverlap`, `span`, and (b) `pulse` of OPL language using sequence of patterns Π_B = {P5,P1,P2,P6,P3,P7,P4}.

The analysis of Fig. 3 shows that the reverse sequence of any sequence of patterns for the MOSP generates the same maximum number of simultaneously open stacks. We can eliminate reverse sequences of patterns in Model MOSP-CP by easily defining that the possible end value of a $pattern_k$, $k \in J$, is $\lfloor N/2 \rfloor + 1$, instead of N. We add one unit here, because each pattern has a fixed size of one instant. In our computational experiments, we chose $k = \arg\max_{j \in J}\{\sum_{i \in I} a_{ij}\}$.

5. Computational experiments

In this section, we present the results of computational experiments performed with the proposed Formulations MOSP-ILP-I and MOSP-ILP-II, and Model MOSP-CP for the MOSP. For benchmark purposes, we consider the other three ILP formulations for the MOSP in the literature, namely Yanasse (1997b), Baptiste (2005), and Lopes & De Carvalho (2015). This section is divided into three parts, and the benchmark instances used in the experiments are presented at the beginning of these sections. We briefly report these three sets of instances in Table 4. All the proposed and benchmark models were coded in C++, and all experiments were carried out on a PC with Intel Xeon E5-2680v2 (2.8 GHz), limited to a single thread, 16 GB of RAM, under a CentOS Linux 7.2.1511 Operating System. We used the IBM CPLEX Optimization Studio v.12.10, as a general-purpose ILP and CP solver. We limited the execution of the solver to 3600 seconds, and in the tables we use the letters “tl” to indicate that this time limit was reached for a given instance or class of instances.

5.1. Results for the set of instances #A

The set of instances #A is composed of 20 instances proposed in Smith & Gent (2005), namely, GP1-8 (N = 50, 100 and M = 50, 100) instances, NWRS1-8 (N = 20, 25, 30, 59 and

Table 4: Sets of benchmark instances.

Set	Number of instances	Source
#A	20 instances	Smith & Gent (2005)
#B	24 instances	Scoop Project (Available in: `http://www.scoop-project.net`)
#C	300 instances	Faggioli & Bentivoglio (1998)

*: all these instances are available in the supplementary material of Frinhani et al. (2018)

$M = 10, 15, 20, 25$) instances, and SP1-4 ($N = 25, 50, 75, 100$ and $M = 25, 50, 75, 100$) instances. Beyond the dimension $M \times N$ of matrix A, the difficulty of solving a MOSP instance increases when the value of the density of matrix A decreases; in other words, in a hard instance of the MOSP, there are several patterns, and each of them produces a few items. The average density of matrix A is 0.673 in GP1-8 instances, 0.232 in NWRS1-8 instances, and 0.069 in SP1-4 instances.

In Table 5, we report results for the set of instances #A. For each instance in this set, we present its name in column Instance, and parameters N, M and the value of the optimal solution in column Nx$M(OPT)$. The optimal values OPT were taken from Frinhani et al. (2018), who made use of the algorithm of Chu & Stuckey (2009). For each of the proposed and benchmark models, we also report the objective function value of the best integer solution found by the solver (column OFV), the lower bound of the linear relaxation (column LP), the relative optimality gap in percentage ($gap[\%]$), and the processing time in seconds ($time[s]$) in Table 5. The relative optimality gap is calculated as $100 \times (OFV - OPT)/(OFV)$, which may not be optimal if the time limit is reached. Note that $gap = 0.0\%$ when the time limit is reached means that an optimal solution was found by the solver, but the certificate of its optimality was not obtained within this interval of time.

The results in Table 5 show that the small values of the average optimality gap were obtained in NWRS1-8 instances, followed by GP1-8 instances, and then SP1-4 instances. Note that NWRS1-8 are the smaller instances in set #A, while GP1-8 (resp. SP1-4) are the most dense (resp. sparse) instances. The solver was able to find an optimal solution and prove its optimality in only 2 (out of 20) instances with the model of Yanasse (1997b), in 7 instances with the model of Lopes & De Carvalho (2015), in 8 instances with MOSP-ILP-II, in 11 instances with the model of Baptiste (2005), in 15 with Formulation MOSP-ILP-I, and in 18 instances with Model MOSP-CP. The average processing time of the solver, considering all instances in this table, was 3,269.46 seconds with the model of Yanasse (1997b), 2,601.00 seconds with the model of Lopes & De Carvalho (2015), 2,523.90 seconds with MOSP-ILP-II, 1,951.11 seconds with the model of Baptiste (2005), 913.41 seconds with Formulation MOSP-ILP-I, and 423.98 seconds with Model MOSP-CP. Although the time limit was reached, we highlight the quality of the solutions obtained by the solver with Model MOSP-CP for SP3 and SP4 instances ($gap < 3.00\%$).

One of the main difficulties of ILP formulations for min-max problems, like the MOSP with its several alternative solutions, is to provide tight lower bounds that can be used to prune non-promising nodes in the context of branch-and-bound algorithms, thus accelerating their convergence. We note the quality of the lower bounds provided by the model of Baptiste (2005) for GP1-8 instances, the model of Lopes & De Carvalho (2015) for NWRS1-8 instances, and Formulation MOSP-ILP-I for GP1-8 and NWRS1-8 instances. In particular, the lower bound provided by the model of Baptiste (2005) for GP5 instance matches the optimal value of this instance; so, too, with Formulation MOSP-ILP-I for GP3, GP6, NWRS1, and NWRS4 instances. In contrast, the weakest lower bounds were provided by the model of Yanasse (1997b), which explain the time limit

Table 5: Results for the set of instances #A.

		Yanasse (1997b)				Baptiste (2005)				Lopes & De Carvalho (2015)			
Instance	Nx$M(OPT)$	OFV	LP	gap[%]	time[s]	OFV	LP	gap[%]	time[s]	OFV	LP	gap[%]	time[s]
GP1	50x50(45)	45	1.00	0.00	tl	45	44.05	0.00	14.00	45	25.02	0.00	tl
GP2	50x50(40)	50	1.00	20.00	tl	40	38.11	0.00	865.19	45	24.02	11.11	tl
GP3	50x50(40)	47	1.00	14.89	tl	40	38.19	0.00	59.63	40	24.38	0.00	tl
GP4	50x50(30)	32	1.00	6.25	tl	30	28.25	0.00	146.12	48	21.08	37.50	tl
GP5	100x100(95)	100	1.00	5.00	tl	95	95.00	0.00	120.15	100	50.23	5.00	tl
GP6	100x100(75)	100	1.00	25.00	tl	100	70.28	25.00	tl	100	0.09	25.00	tl
GP7	100x100(75)	100	1.00	25.00	tl	75	71.21	0.00	3,156.79	100	47.18	25.00	tl
GP8	100x100(60)	100	1.00	40.00	tl	100	56.35	40.00	tl	100	0.57	40.00	tl
Avg	**(57.50)**	**71.75**	**1.00**	**17.02**	**3600.00**	**65.63**	**55.18**	**8.13**	**1445.24**	**72.25**	**24.07**	**17.95**	**3600.00**
NWRS1	20x10(3)	3	0.50	0.00	569.05	3	0.59	0.00	1.26	3	2.78	0.00	0.05
NWRS2	20x10(4)	4	0.50	0.00	20.17	4	0.62	0.00	1.50	4	3.24	0.00	0.05
NWRS3	25x15(7)	7	0.60	0.00	tl	7	0.78	0.00	145.14	7	4.59	0.00	4.57
NWRS4	25x15(7)	7	0.60	0.00	tl	7	0.76	0.00	36.78	7	5.28	0.00	2.06
NWRS5	30x20(12)	12	0.67	0.00	tl	12	0.94	0.00	tl	12	8.06	0.00	1,067.43
NWRS6	30x20(12)	12	0.67	0.00	tl	12	0.94	0.00	2,075.59	12	8.15	0.00	2,168.10
NWRS7	59x25(10)	13	0.42	23.08	tl	12	0.56	16.67	tl	10	6.96	0.00	1,977.77
NWRS8	59x25(16)	20	0.42	20.00	tl	16	0.59	0.00	tl	16	9.57	0.00	tl
Avg	**(8.88)**	**9.75**	**0.55**	**5.38**	**2773.65**	**9.13**	**0.72**	**2.08**	**1632.53**	**8.88**	**6.08**	**0.00**	**1102.51**
SP1	25x25(9)	10	1.00	10.00	tl	9	1.19	0.00	tl	9	5.15	0.00	tl
SP2	50x50(19)	25	1.00	24.00	tl	24	1.13	20.83	tl	33	8.32	42.42	tl
SP3	75x75(34)	73	1.00	53.42	tl	75	1.10	54.67	tl	75	11.41	54.67	tl
SP4	100x100(53)	100	1.00	47.00	tl	100	1.00	47.00	tl	100	0.00	47.00	tl
Avg	**(28.75)**	**32.84**	**0.76**	**22.86**	**3600.00**	**32.14**	**0.90**	**17.66**	**3600.00**	**32.98**	**6.95**	**18.01**	**3600.00**

		Formulation MOSP-ILP-I				Formulation MOSP-ILP-II				Model MOSP-CP		
		OFV	LP	gap[%]	time[s]	OFV	LP	gap[%]	time[s]	OFV	gap[%]	time[s]
GP1	50x50(45)	45	40.00	0.00	0.46	45	35.66	0.00	8.06	45	0.00	0.07
GP2	50x50(40)	40	33.00	0.00	4.00	40	26.52	0.00	911.64	40	0.00	1.96
GP3	50x50(40)	40	40.00	0.00	0.51	42	28.50	4.76	tl	40	0.00	1.02
GP4	50x50(30)	30	25.33	0.00	20.18	30	22.17	0.00	95.17	30	0.00	3.46
GP5	100x100(95)	95	90.67	0.00	1.79	95	79.00	0.00	2,687.13	95	0.00	0.64
GP6	100x100(75)	75	75.00	0.00	22.47	89	62.00	15.73	tl	75	0.00	10.40
GP7	100x100(75)	75	71.67	0.00	194.33	78	59.44	3.85	tl	75	0.00	147.79
GP8	100x100(60)	61	54.00	1.64	tl	98	49.00	38.78	tl	60	0.00	254.81
Avg	**(57.50)**	**57.63**	**53.71**	**0.20**	**480.47**	**64.63**	**45.29**	**7.89**	**2262.75**	**57.50**	**0.00**	**52.52**
NWRS1	20x10(3)	3	3.00	0.00	0.04	3	1.00	0.00	17.01	3	0.00	0.01
NWRS2	20x10(4)	4	3.00	0.00	0.05	4	2.00	0.00	3.82	4	0.00	0.08
NWRS3	25x15(7)	7	5.00	0.00	0.37	7	1.20	0.00	3,518.35	7	0.00	1.02
NWRS4	25x15(7)	7	7.00	0.00	0.17	7	2.00	0.00	36.87	7	0.00	0.88
NWRS5	30x20(12)	12	10.00	0.00	0.60	12	1.53	0.00	tl	12	0.00	3.78
NWRS6	30x20(12)	12	11.00	0.00	0.25	12	2.00	0.00	tl	12	0.00	2.63
NWRS7	59x25(10)	10	6.00	0.00	18.21	11	1.00	9.09	tl	10	0.00	72.49
NWRS8	59x25(16)	16	12.00	0.00	4.76	16	1.24	0.00	tl	16	0.00	103.22
Avg	**(8.88)**	**8.88**	**7.13**	**0.00**	**3.06**	**9.00**	**1.50**	**1.14**	**2247.01**	**8.88**	**0.00**	**23.02**
SP1	25x25(9)	9	2.00	0.00	tl	9	1.05	0.00	tl	9	0.00	3.19
SP2	50x50(19)	23	2.00	17.39	tl	22	1.03	13.64	tl	19	0.00	672.23
SP3	75x75(34)	67	5.00	49.25	tl	39	1.22	12.82	tl	35	2.86	tl
SP4	100x100(53)	89	4.00	40.45	tl	69	1.05	23.19	tl	53	0.00	tl
Avg	**(28.75)**	**29.36**	**6.14**	**13.39**	**3600.00**	**23.38**	**1.26**	**7.48**	**3600.00**	**20.36**	**0.36**	**1968.85**

reached in 18 (out of 20) instances; therefore, we disregard this model in the following analysis.

5.2. Results for the set of instances #B

The set of instances #B is composed of 24 real-life instances, proposed in the context of the Scoop Project. These instances are divided into two groups, namely companies A and B. We have parameter N varying from 13 to 37, and parameter M varying from 20 to 134 for the instances of company A; and parameter N from 10 to 49, and parameter M from 13 to 60 for the instances of company B. The average density of matrix A is 0.080 in the instances of company A, and 0.171 in the instances of company B.

In Table 6, we report results for the set of instances #B. The results show that the small values of the average optimality gap and processing time were obtained in the instances of company B in comparison with the instances of company A, which can be explained by the average dimension of the instances and their densities. The solver was able to find an optimal solution and prove its optimality in 7 (out of 24) instances with MOSP-ILP-II, 10 with Formulation MOSP-ILP-I, 11 instances with the model of Baptiste (2005), 12 instances with the model of Lopes & De Carvalho (2015), and 23 instances with Model MOSP-CP. The average processing time of the solver, considering all instances in this table, was 2,562.54 seconds with MOSP-ILP-II, 2,140.78 seconds with Formulation MOSP-ILP-I, 2,058.45 seconds with the model of Baptiste (2005), 2,042.57 seconds with the model of Lopes & De Carvalho (2015), and 254.22 seconds with Model MOSP-CP.

The number of items is always greater than the number of patterns in the instances of set #B (i.e., $M > N$), in contrast to the instances of set #A where $M \leq N$. In general, sequencing the patterns seems to be a better idea when $M > N$; conversely, sequencing the closed stacks seems to be appropriate when $M < N$, or when the density of the MOSP graph is considerably greater than the density of matrix A. In this context, for the instances of company A, note that the performance of the solver with the approaches that sequence the closed stacks (i.e., the model of Lopes & De Carvalho (2015) and Formulation MOSP-ILP-I) is significantly worse than with the approaches that sequence the patterns (i.e., the model of Baptiste (2005), MOSP-ILP-II and Model MOSP-CP). See the first "Avg" line in Table 6).

5.3. Results for the set of instances #C

The set of instances #C is composed of 300 random instances, proposed in Faggioli & Bentivoglio (1998), with $N = 10, 20, 30, 40, 50$ and $M = 10, 15, 20, 25, 30, 40$. Each combination of $N \times M$ represents a class of instances, and has 10 instances ($300 = 5 \times 6 \times 10$). The average density of matrix A is 0.159 in these instances with a standard deviation of 0.083.

In Table 7, we report results for the set of instances #C. Each row in this table presents results for a combination $N \times M$ of 10 instances. For each proposed and benchmark approach, we also report the number of proven optimal solutions in column #O. For every approach and class of instances, if $gap = 0.0\%$ and #O< 10, then an optimal solution was found by the solver in each of the 10 instances; but the certificates of optimality were not obtained for all these instances within this interval of time. The solver was able to find an optimal solution and prove its optimality in 104 (out of 300) instances with the model of Lopes & De Carvalho (2015), 118 with MOSP-ILP-II, 126 with the model of Baptiste (2005), 136 with Formulation MOSP-ILP-I, and in all 300 instances with Model MOSP-CP. The average processing time of the solver, considering all instances in this table, was 2,490.87 seconds with the model of Lopes & De Carvalho (2015), 2,342.79 seconds with MOSP-ILP-II, 2,294.90 seconds with the model of Baptiste (2005), 2,079.93 seconds with Formulation MOSP-ILP-I, and only 10.75 seconds with Model MOSP-CP.

The solver reached the time limit in all classes of instances with $M = 40, 50$ items with the model of Lopes & De Carvalho (2015) and Formulation MOSP-ILP-I, the approaches that sequence the closed stacks, even when the number of patterns was small. In contrast, the performance of the solver with these two approaches was better than with the model of Baptiste (2005) and MOSP-ILP-II, approaches that sequence the patterns, in the classes of $N = 40$ patterns and $M = 10, 20$ items.

Table 6: Results for the set of instances #B.

		Baptiste (2005)				Lopes & De Carvalho (2015)				Formulation MOSP-ILP-I				Formulation MOSP-ILP-II				Model MOSP-CP		
Instance	Nx$M(OPT)$	OFV	LP	gap[%]	time[s]	OFV	LP	gap[%]	time[s]	OFV	LP	gap[%]	time[s]	OFV	LP	gap[%]	time[s]	OFV	gap[%]	time[s]
A_AP-9.d_10	13x20 (6)	6	1.76	0.00	8.46	6	3.67	0.00	71.24	6	2.00	0.00	44.97	6	2.00	0.00	103.94	6	0.00	0.47
A_AP-9.d_3	16x20 (6)	6	1.33	0.00	2,301.72	6	3.23	0.00	168.53	6	3.00	0.00	399.56	6	2.00	0.00	tl	6	0.00	1.10
A_FA+AA-_15	18x68 (9)	9	3.86	0.00	tl	68	5.28	86.76	tl	14	3.00	35.71	tl	9	3.79	0.00	tl	9	0.00	5.00
A_FA+AA-_2	19x75 (11)	11	4.07	0.00	tl	75	0.00	85.33	tl	13	4.00	15.38	tl	11	4.05	0.00	tl	11	0.00	4.73
A_AP-9.d_6	20x31 (5)	5	1.60	0.00	tl	5	2.92	0.00	tl	5	1.00	0.00	tl	5	1.55	0.00	tl	5	0.00	1.70
A_FA+AA-_12	20x75 (9)	9	3.83	0.00	tl	75	0.00	88.00	tl	11	3.00	18.18	tl	9	3.83	0.00	tl	9	0.00	5.44
A_AP-9.d_11	21x27 (6)	6	1.38	0.00	tl	6	3.40	0.00	tl	6	2.00	0.00	tl	6	1.37	0.00	tl	6	0.00	5.93
A_FA+AA-_6	21x79 (13)	14	3.94	7.14	tl	79	0.00	83.54	tl	18	3.00	27.78	tl	14	3.83	7.14	tl	13	0.00	46.80
A_FA+AA-_8	28x82 (11)	13	3.06	15.38	tl	82	0.00	86.59	tl	19	2.00	42.11	tl	13	2.98	15.38	tl	11	0.00	52.36
A_FA+AA-_11	28x99 (11)	12	3.62	8.33	tl	99	0.00	88.89	tl	16	2.00	31.25	tl	11	3.55	0.00	tl	11	0.00	238.53
A_FA+AA-_1	37x105 (12)	15	2.94	20.00	tl	105	0.00	88.57	tl	19	2.00	36.84	tl	14	2.86	14.29	tl	12	0.00	2,073.81
A_FA+AA-_13	37x134 (17)	23	3.73	26.09	tl	134	0.00	87.31	tl	25	3.00	32.00	tl	21	3.66	19.05	tl	17	0.00	tl
Avg	**(8.13)**	**10.75**	**2.93**	**6.41**	**3192.52**	**61.67**	**1.54**	**57.92**	**3019.98**	**13.17**	**2.50**	**19.94**	**3037.04**	**10.42**	**2.96**	**4.66**	**3308.66**	**9.67**	**0.00**	**502.99**
B_22X18_50	10x14 (10)	10	1.91	0.00	2.68	10	5.39	0.00	135.61	10	6.00	0.00	0.29	10	3.00	0.00	8.21	10	0.00	0.25
B_39Q18_82	10x14 (5)	5	2.41	0.00	0.20	5	3.47	0.00	0.69	5	4.00	0.00	0.25	5	3.00	0.00	0.29	5	0.00	0.01
B_42F22_93	10x17 (5)	5	1.87	0.00	1.35	5	3.22	0.00	2.22	5	3.00	0.00	18.48	5	1.89	0.00	1.62	5	0.00	0.21
B_18AB1_32	11x14 (6)	6	2.00	0.00	0.96	6	4.13	0.00	1.60	6	3.00	0.00	0.42	6	1.63	0.00	0.89	6	0.00	0.11
B_CARLET_137	12x13 (5)	5	1.29	0.00	0.44	5	3.56	0.00	0.63	5	4.00	0.00	0.07	5	2.50	0.00	1.51	5	0.00	0.03
B_12F18_11	15x21 (6)	6	1.55	0.00	23.88	6	3.67	0.00	307.10	6	3.00	0.00	271.88	6	2.00	0.00	tl	6	0.00	0.37
B_18CR1_33	18x19 (4)	4	1.08	0.00	115.76	4	2.62	0.00	20.29	4	1.00	0.00	136.76	4	1.09	0.00	184.57	4	0.00	0.35
B_GTM18A_139	20x24 (5)	5	1.24	0.00	tl	5	3.03	0.00	1,307.92	5	1.00	0.00	tl	5	1.50	0.00	tl	5	0.00	2.83
B_23B25_52	21x27 (5)	5	1.29	0.00	tl	5	3.03	0.00	2,418.67	5	2.00	0.00	tl	5	1.50	0.00	tl	5	0.00	0.97
B_12M18_12	22x28 (6)	6	1.40	0.00	48.82	6	4.02	0.00	tl	6	2.00	0.00	105.93	6	2.00	0.00	tl	6	0.00	1.67
B_CUC28A_138	26x31 (6)	6	1.23	0.00	98.43	6	4.05	0.00	1,387.21	6	1.00	0.00	tl	6	1.33	0.00	tl	6	0.00	0.93
B_REVAL_145	49x60 (7)	9	1.26	22.22	tl	7	0.00	0.00	tl	9	2.00	22.22	tl	7	1.50	0.00	tl	7	0.00	57.69
Avg	**(5.75)**	**6.00**	**1.54**	**1.85**	**924.38**	**5.83**	**3.35**	**0.00**	**1065.16**	**6.00**	**2.67**	**1.85**	**1244.51**	**5.83**	**1.91**	**0.00**	**1816.42**	**5.83**	**0.00**	**5.45**

Table 7: Results for the set of instances #C.

	Baptiste (2005)				Lopes & De Carvalho (2015)				Formulation MOSP-ILP-I				Formulation MOSP-ILP-II				Model MOSP-CP			
NxM (OPT)	OFV	gap[%]	time[s]	#O	OFV	gap[%]	time[s]	#O	OFV	gap[%]	time[s]	#O	OFV	gap[%]	time[s]	#O	OFV	gap[%]	time[s]	#O
10x10 (5.50)	5.50	0.00	0.33	10	5.50	0.00	0.15	10	5.50	0.00	0.03	10	5.50	0.00	0.81	10	5.50	0.00	0.08	10
10x20 (6.20)	6.20	0.00	1.89	10	6.20	0.00	309.02	10	6.20	0.00	32.85	10	6.20	0.00	3.49	10	6.20	0.00	0.26	10
10x30 (6.10)	6.10	0.00	2.72	10	6.20	1.61	3,369.33	1	6.30	3.17	3,023.17	4	6.10	0.00	1.82	10	6.10	0.00	0.20	10
10x40 (7.70)	7.70	0.00	8.08	10	8.50	9.41	tl	0	7.90	2.53	tl	0	7.70	0.00	4.04	10	7.70	0.00	0.68	10
10x50 (8.20)	8.20	0.00	2.20	10	9.30	11.83	tl	0	9.00	8.89	tl	0	8.20	0.00	0.74	10	8.20	0.00	0.40	10
15x10 (6.60)	6.60	0.00	3.10	10	6.60	0.00	0.28	10	6.60	0.00	0.03	10	6.60	0.00	35.66	10	6.60	0.00	0.34	10
15x20 (7.20)	7.20	0.00	871.37	9	7.20	0.00	346.63	10	7.20	0.00	37.33	10	7.20	0.00	392.04	10	7.20	0.00	0.74	10
15x30 (7.30)	7.30	0.00	1,524.62	8	7.70	5.19	3,244.06	2	7.50	2.67	3,432.36	1	7.30	0.00	916.45	9	7.30	0.00	1.61	10
15x40 (7.20)	7.20	0.00	1,893.60	8	11.20	35.71	tl	0	8.10	11.11	tl	0	7.20	0.00	1,766.20	7	7.20	0.00	2.17	10
15x50 (7.40)	7.40	0.00	2,259.24	4	28.10	73.67	tl	0	8.30	10.84	tl	0	7.40	0.00	1,839.23	6	7.40	0.00	3.02	10
20x10 (7.50)	7.50	0.00	126.28	10	7.50	0.00	0.33	10	7.50	0.00	0.02	10	7.50	0.00	690.50	10	7.50	0.00	0.46	10
20x20 (8.50)	8.50	0.00	3,122.76	4	8.50	0.00	768.07	10	8.50	0.00	10.91	10	8.50	0.00	2,454.16	5	8.50	0.00	1.48	10
20x30 (8.80)	9.20	4.35	tl	0	9.30	5.38	tl	0	9.10	3.30	3,366.27	3	8.90	1.12	3,331.72	1	8.80	0.00	2.83	10
20x40 (8.50)	8.80	3.41	tl	0	12.90	34.11	tl	0	9.70	12.37	tl	0	8.70	2.30	tl	0	8.50	0.00	6.21	10
20x50 (7.90)	8.30	4.82	tl	0	29.50	73.22	tl	0	9.20	14.13	tl	0	8.10	2.47	3,267.51	1	7.90	0.00	10.14	10
25x10 (8.00)	8.00	0.00	1,087.55	10	8.00	0.00	0.44	10	8.00	0.00	0.02	10	8.00	0.00	2,772.67	4	8.00	0.00	0.86	10
25x20 (9.80)	9.90	1.01	tl	0	9.80	0.00	2,458.67	7	9.80	0.00	17.15	10	9.90	1.01	3,470.84	1	9.80	0.00	2.18	10
25x30 (10.50)	10.90	3.67	tl	0	10.90	3.67	tl	0	10.60	0.94	3,095.20	2	10.80	2.78	tl	0	10.50	0.00	4.02	10
25x40 (10.30)	11.00	6.36	tl	0	25.10	58.96	tl	0	11.50	10.43	tl	0	10.80	4.63	tl	0	10.30	0.00	11.25	10
25x50 (10.00)	10.90	8.26	tl	0	45.70	78.12	tl	0	12.30	18.70	tl	0	10.60	5.66	tl	0	10.00	0.00	16.98	10
30x10 (7.80)	7.80	0.00	1,215.23	9	7.80	0.00	0.19	10	7.80	0.00	0.02	10	7.80	0.00	2,535.88	4	7.80	0.00	1.25	10
30x20 (11.10)	11.40	2.63	tl	0	11.10	0.00	3,028.73	4	11.10	0.00	14.03	10	11.40	2.63	tl	0	11.10	0.00	3.22	10
30x30 (12.20)	12.90	5.43	tl	0	12.40	1.61	tl	0	12.20	0.00	2,785.27	4	12.70	3.94	tl	0	12.20	0.00	7.04	10
30x40 (12.10)	13.10	7.63	tl	0	28.00	56.79	tl	0	12.80	5.47	tl	0	13.20	8.33	tl	0	12.10	0.00	19.27	10
30x50 (11.20)	12.60	11.11	tl	0	43.70	74.37	tl	0	13.90	19.42	tl	0	12.00	6.67	tl	0	11.20	0.00	24.36	10
40x10 (8.40)	8.40	0.00	2,728.14	4	8.40	0.00	0.28	10	8.40	0.00	0.02	10	8.40	0.00	tl	0	8.40	0.00	2.05	10
40x20 (13.00)	13.50	3.70	tl	0	13.00	0.00	tl	0	13.00	0.00	6.59	10	13.40	2.99	tl	0	13.00	0.00	8.44	10
40x30 (14.50)	16.20	10.49	tl	0	14.80	2.03	tl	0	14.50	0.00	3,376.55	2	15.60	7.05	tl	0	14.50	0.00	24.49	10
40x40 (14.90)	16.60	10.24	tl	0	18.70	20.32	tl	0	15.90	6.29	tl	0	16.70	10.78	tl	0	14.90	0.00	37.72	10
40x50 (14.60)	18.20	19.78	tl	0	43.30	66.28	tl	0	18.10	19.34	tl	0	17.10	14.62	tl	0	14.60	0.00	128.78	10
Sum/Avg (279)	**293.10**	**3.43**	**2,294.90**	**126**	**464.90**	**20.41**	**2,490.87**	**104**	**296.50**	**4.99**	**2,079.93**	**136**	**289.50**	**2.57**	**2,342.79**	**118**	**279.00**	**0.00**	**10.75**	**300**

*: each row in this table presents results for a class of 10 instances.

**: the last row presents summations in columns OFV and #O, and averages in columns gap[%] and time[s].

In summary, the main results of these sections point out that: (a) sequencing the patterns seems to be a better idea when $M > N$, and sequencing the closed stacks seems to be appropriate when $M < N$; (b) the ILP formulations provided tight lower bounds for a small class of problem instances, which may explain the large values of processing time; and, (c) Model MOSP-CP outperformed all the ILP formulations, both in quality of solution and processing time. Notably, the solver achieved optimality with the MOSP-CP in all 300 instances, with an average processing time of only 10.75 seconds. Rhe second best average processing time was of Formulation MOSP-ILP-I at 2,079.83 seconds and only produced 136 proven optimal solutions.

6. Conclusions

We addressed the Minimization of Open Stacks Problem (MOSP). Real-life applications of the MOSP include the sequencing of patterns in manufacturing industries, such as in the cutting of objects in the glass, furniture, and metallurgical industries to comply with space requirements around the cutting machine, and also in the field of VLSI design for obtaining small integrated circuits. The MOSP has been mainly addressed by heuristics/meta-heuristics (Yuen, 1991; Becceneri et al., 2004; Carvalho & Soma, 2015; Gonçalves et al., 2016; Frinhani et al., 2018). To the best of our knowledge, the only Integer Linear Programming (ILP) formulations for the MOSP in the literature correspond to the approaches of Yanasse (1997b), Baptiste (2005) and Lopes & De Carvalho (2015), but these studies do not analyze the relative computational performance of these formulations.

In this sense, we presented a novel ILP formulation for the MOSP. The insight of this approach is to sequence the stacks to be closed, inspired by the MOSP graph, where a node of the graph is closed, then all its adjacent nodes have been opened. We derived an alternative ILP formulation for the problem sequencing the patterns from the approach of Faggioli & Bentivoglio (1998). We also developed a simple Constraint Programming (CP) model to sequence the patterns, which mades use of interval variables for modeling the patterns and items, and the idea of renewable resources for the stacks. Computational experiments performed with benchmark instances showed that the proposed CP model outperformed the proposed and benchmark ILP formulations in quality of solution and processing times, in the context of small and moderate sized problem instances. The proposed ILP formulations excelled in instances characterized by a dense problem instance/MOSP graph.

As future research, the ILP formulations could be enhanced with the proposition of new valid inequalities to strengthen their linear relaxation. Since these mathematical formulations have barely been explored, another possibility is to develop solution methods based on decomposition methods for large scale optimization problems, like Dantzig-Wolfe decomposition, Benders decomposition, or Lagrangian relaxation, which could enable the solution of larger problem instances and/or contribute to obtaining more certificates of optimality. An interesting possibility would be the integration of the MOSP with decisions of pattern generation and pattern minimization problems. This integration may help eliminate symmetrical solutions of the MOSP.

Acknowledgements

The authors would like to thank the São Paulo Research Foundation (FAPESP-Brazil) [grant numbers 2020/00747-2, 2016/01860-1] and the National Council for Scientific and Technological

Development (CNPq-Brazil) [grant number 304144/2019-3] for their financial support. The research was carried out, using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP-Brazil [grant number 2013/07375-0].

References

Armbruster, M. (2002). A solution procedure for a pattern sequencing problem as part of a one-dimensional cutting stock problem in the steel industry. *European Journal of Operational Research*, *141*, 328–340. doi:10.1016/S0377-2217(02)00128-5.

Ashikaga, F. M., & Soma, N. Y. (2009). A heuristic for the minimization of open stacks problem. *Pesquisa Operacional*, *29*, 439–450. doi:10.1590/S0101-74382009000200010.

Baptiste, P. (2005). Simple MIP formulations to minimize the maximum number of open stacks. In B. M. Smith, & I. P. Gent (Eds.), *Constraint Modelling Challenge 2005*. Edinburgh. URL: https://ipg.host.cs.st-andrews.ac.uk/challenge/.

Becceneri, J. C., Yanasse, H. H., & Soma, N. Y. (2004). A method for solving the minimization of the maximum number of open stacks problem within a cutting process. *Computers and Operations Research*, *31*, 2315–2332. doi:10.1016/S0305-0548(03)00189-8.

Belov, G., & Scheithauer, G. (2007). Setup and open-stacks minimization in one-dimensional stock cutting. *INFORMS Journal on Computing*, *19*, 27–35. doi:10.1287/ijoc.1050.0132.

Carvalho, M. A. M., & Soma, N. Y. (2015). A breadth-first search applied to the minimization of the open stacks. *Journal of the Operational Research Society*, *66*, 936–946. doi:10.1057/jors.2014.60.

Chu, G., & Stuckey, P. J. (2009). Minimizing the maximum number of open stacks by customer search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5732 LNCS*, 242–257. doi:10.1007/978-3-642-04244-7_21.

Costa, R., Frinhani, R. M. D., & Soma, N. Y. (2017). Evaluation of use of sequencing algorithm in the optimization of the assembly process of automotive vehicles: a case study (in portuguese language). In *Annals of XVIII Oficina Nacional de Problemas de Corte e Empacotamento, Planejamento e Programação de Produção e Correlatos, Universidade Federal de São Paulo (UNIFESP), São José dos Campos, Brazil*.

Cui, Y., Zhong, C., & Yao, Y. (2015). Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost. *European Journal of Operational Research*, *243*, 540 – 546. doi:10.1016/j.ejor.2014.12.015.

De Giovanni, L., Massi, G., & Pezzella, F. (2013a). An adaptive genetic algorithm for large-size open stack problems. *International Journal of Production Research*, *51*, 682–697. doi:10.1080/00207543.2012.657256.

De Giovanni, L., Massi, G., Pezzella, F., Pfetsch, M. E., Rinaldi, G., & Ventura, P. (2013b). A heuristic and an exact method for the gate matrix connection cost minimization problem. *International Transactions in Operational Research*, *20*, 627–643. doi:10.1111/itor.12025.

De La Banda, M. G., & Stuckey, P. J. (2007). Dynamic programming to minimize the maximum number of open stacks. *INFORMS Journal on Computing*, *19*, 607–617. doi:10.1287/ijoc.1060.0205.

De Oliveira, A. C., & Lorena, L. A. (2002). A constructive genetic algorithm for gate matrix layout problems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *21*, 969–974. doi:10.1109/TCAD.2002.800454.

Deo, N., Krishnamoorthy, M. S., & Langston, M. A. (1987). Exact and Approximate Solutions for the Gate Matrix Layout Problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *6*, 79–84. doi:10.1109/TCAD.1987.1270248.

Dyson, R. G., & Gregory, A. S. (1974). The Cutting Stock Problem in the Flat Glass Industry. *Journal of the Operational Research Society*, *25*, 41–53. doi:10.1057/jors.1974.5.

Faggioli, E., & Bentivoglio, C. A. (1998). Heuristic and exact methods for the cutting sequencing problem. *European Journal of Operational Research*, *110*, 564–575. doi:10.1016/S0377-2217(97)00268-3.

Fink, A., & Voß, S. (1999). Applications of modern heuristic search methods to pattern sequencing problems. *Computers and Operations Research*, *26*, 17–34. doi:10.1016/S0305-0548(98)80001-4.

Foerster, H., & Wäscher, G. (1998). Simulated annealing for order spread minimization in sequencing cutting patterns. *European Journal of Operational Research*, *110*, 272–281. doi:10.1016/S0377-2217(97)00257-9.

Frinhani, R. M. D., Carvalho, M. A. M., & Soma, N. Y. (2018). A PageRank-based heuristic for the minimization of open stacks problem. *PLoS ONE*, *13*, 1–24. doi:10.1371/journal.pone.0203076.

Gonçalves, J. F., Resende, M. G., & Costa, M. D. (2016). A biased random-key genetic algorithm for the minimization of open stacks problem. *International Transactions in Operational Research*, *23*, 25–46. doi:10.1111/itor.12109.

Lima, J. R., & Carvalho, M. A. M. (2017). Descent search approaches applied to the minimization of open stacks. *Computers and Industrial Engineering*, *112*, 175–186. doi:10.1016/j.cie.2017.08.016.

Linhares, A., & Yanasse, H. H. (2002). Connections between cutting-pattern sequencing, VLSI design, and flexible machines. *Computers and Operations Research*, *29*, 1759–1772. doi:10.1016/S0305-0548(01)00054-5.

Linhares, A., Yanasse, H. H., & Torreão, J. R. (1999). Linear gate assignment: A fast statistical mechanics approach. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *18*, 1750–1758. doi:10.1109/43.811324.

Lopes, I. C., & De Carvalho, J. M. (2015). Graph properties of minimization of open stacks problems and a new integer programming model. *Pesquisa Operacional*, *35*, 213–250. doi:10.1590/0101-7438.2015.035.02.0213.

Madsen, O. B. G. (1979). Glass cutting in a small firm. *Mathematical Programming*, *17*, 85–90. doi:10.1007/BF01588227.

Martin, M., Moretti, A., Gomes-Ruggiero, M., & Neto, L. S. (2018). Modification of Haessler's sequential heuristic procedure for the one-dimensional cutting stock problem with setup cost. *Production*, *28*. doi:10.1590/0103-6513.20170105.

Matsumoto, K., Umetani, S., & Nagamochi, H. (2011). On the one-dimensional stock cutting problem in the paper tube industry. *Journal of Scheduling*, *14*, 281–290. doi:10.1007/s10951-010-0164-2.

Rinaldi, F., & Franz, A. (2007). A two-dimensional strip cutting problem with sequencing constraint. *European Journal of Operational Research*, *183*, 1371–1384. doi:10.1016/j.ejor.2005.12.050.

Santos, V. G. M., & Carvalho, M. A. M. (2018). Adaptive large neighborhood search applied to the design of electronic circuits. *Applied Soft Computing Journal*, *73*, 14–23. doi:10.1016/j.asoc.2018.08.017.

Silva, T. T., Chaves, A. A., & Yanasse, H. H. (2020). A new multicommodity flow model for the job sequencing and tool switching problem. *International Journal of Production Research*, *7543*, 1–16. doi:10.1080/00207543.2020.1748906.

Smith, B. M., & Gent, I. P. (2005). Constraint modelling challenge 2005. In *The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 05, Edinburgh*. URL: https://ipg.host.cs.st-andrews.ac.uk/challenge/.

Soares, L. C., Reinsma, J. A., Nascimento, L. H., & Carvalho, M. A. M. (2020). Heuristic methods to consecutive block minimization. *Computers and Operations Research*, *120*, 104948. doi:10.1016/j.cor.2020.104948.

Tang, C. S., & Denardo, E. V. (1988). Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches. *Operations Research*, *36*, 767–777. doi:10.1287/opre.36.5.767.

Van Hentenryck, P. (1999). *The OPL Optimization Programming Language*. Cambridge, MA, USA: MIT Press.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, *183*, 1109–1130. doi:10.1016/j.ejor.2005.12.047.

Yanasse, H. H. (1997a). A transformation for solving a pattern sequencing problem in the wood cut industry. *Pesquisa Operacional*, *17*, 57–70.

Yanasse, H. H. (1997b). On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, *100*, 454–463. doi:10.1016/S0377-2217(97)84107-0.

Yanasse, H. H., Becceneri, J. C., & Soma, N. Y. (1999). Bounds for a problem of sequencing patterns. *Pesquisa Operacional*, *19*, 249–277.

Yanasse, H. H., & Lamosa, M. J. P. (2007). An integrated cutting stock and sequencing problem. *European Journal of Operational Research*, *183*, 1353–1370. doi:10.1016/j.ejor.2005.09.054.

Yanasse, H. H., & Limeira, M. S. (2004). Refinements on an enumeration scheme for solving a pattern sequencing problem. *International Transactions in Operational Research*, *11*, 277–292. doi:10.1111/j.1475-3995.2004.00458.x.

Yanasse, H. H., & Pinto, M. J. (2003). Uma nova formulação para um problema de sequenciamento de padrões em ambientes de corte. In *Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional*. URL: http://www.din.uem.br/sbpo/sbpo2003/pdf/arq0180.pdf.

Yanasse, H. H., & Senne, E. L. F. (2010). The minimization of open stacks problem: A review of some properties and their use in pre-processing operations. *European Journal of Operational Research*, *203*, 559–567. doi:10.1016/j.ejor.2009.09.017.

Yuen, B. J. (1991). Heuristics for sequencing cutting patterns. *European Journal of Operational Research*, *55*, 183–190. doi:10.1016/0377-2217(91)90222-H.

Yuen, B. J. (1995). Improved heuristics for sequencing cutting patterns. *European Journal of Operational Research*, *87*, 57–64. doi:10.1016/0377-2217(94)00068-N.

Yuen, B. J., & Richardson, K. V. (1995). Establishing the optimality of sequencing heuristics for cutting stock problems. *European Journal of Operational Research*, *84*, 590–598. doi:10.1016/0377-2217(95)00025-L.