UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA
MESTRADO EM MODELAGEM E MÉTODOS QUANTITATIVOS

RONALDO LAGE PESSOA

CONTRIBUTIONS TO THE TWO-DIMENSIONAL GUILLOTINE CUTTING STOCK PROBLEM

FORTALEZA
2017

RONALDO LAGE PESSOA

CONTRIBUTIONS TO THE TWO-DIMENSIONAL GUILLOTINE CUTTING STOCK PROBLEM

Dissertação apresentada ao Curso de Mestrado em Modelagem e Métodos Quantitativos do Departamento de Estatística e Matemática Aplicada do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Modelagem e Métodos Quantitativos. Área de Concentração: Otimização e Inteligência Computacional

Orientador: Prof. Dr. Bruno de Athayde Prata

Co-Orientador: Prof. Dr. Carlos Diego Rodrigues

FORTALEZA

2017

RONALDO LAGE PESSOA

CONTRIBUTIONS TO THE TWO-DIMENSIONAL GUILLOTINE CUTTING STOCK PROBLEM

Dissertação apresentada ao Curso de Mestrado em Modelagem e Métodos Quantitativos do Departamento de Estatística e Matemática Aplicada do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Modelagem e Métodos Quantitativos. Área de Concentração: Otimização e Inteligência Computacional

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Bruno de Athayde Prata (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos Diego Rodrigues (Co-Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Albert Einstein F. Muritiba
Universidade Federal do Ceará (UFC)

Prof. Dr. Reinaldo Morabito
Universidade Federal de São Carlos (UFSCar)

AGRADECIMENTOS

Agradeço inicialmente a Deus que, por intermédio de Nossa Senhora, permitiu que o presente trabalho pudesse ser finalizado no prazo e qualidade desejada, mesmo com todas as barreiras pessoais que surgiram durante o seu desenvolvimento.

A minha esposa Larissa, pela paciência, compreensão e o esforço extra nos cuidados do nosso filho, tendo ele nascido durante o desenvolvimento desta pesquisa.

Ao meu filho Miguel, para o qual todo o esforço despendido nesse trabalho foi dedicado.

A CAPES, pelo apoio fornecido através da bolsa de pesquisa, permitindo a dedicação total a esta pesquisa.

Finalmente, agradeço aos meus orientadores Prof. Bruno Prata e Prof. Carlos Diego pelos suportes técnico, científico e motivacional que foram decisivos para o desenvolvimento deste trabalho.

“Não há lugar para a sabedoria onde não há paciência.”

(Santo Agostinho)

RESUMO

No presente trabalho são apresentadas duas novas variantes do problema de corte guilhotinado bi-dimensional. São propostas formulações matemáticas e métodos de solução para lidar com os problemas apresentados. Primeiramente, é apresentado o problema de corte guilhotinado bi-dimensional de dois-estágios no qual os itens são idênticos, as placas têm tamanhos diferentes e o objetivo é determinar o tamanho ótimo dos itens idênticos. Dois procedimentos de solução são apresentados para resolver o caso no qual a orientação dos itens é fixa e o caso no qual a rotação ortogonal dos itens é permitida. Os dois procedimentos lidam com o problema resolvendo iterativamente um problema da mochila para cada tamanho possível de item e retornando a melhor solução encontrada. Experimentos numéricos são conduzidos para avaliar a escalabilidade das abordagens. Por último, é apresentado o problema de corte bi-dimensional guilhotinado de k-estágios no qual o custo de *setup* associado aos estágios de corte é considerado relevante. Uma formulação matemática com $O(n^2pN)$ variáveis e $O(npN)$ restrições é apresentada, onde, n, p e k são o número de itens, placas e estágios, respectivamente. Experimentos computacionais foram conduzidos em vinte instâncias de pequena escala geradas aleatoriamente para avaliar a qualidade da abordagem.

Palavras-chave: Problemas de corte e empacotamento; Itens idênticos; *Setup*; Corte em estágios; Programação matemática.

ABSTRACT

In this work we present two new variants of the two-dimensional guillotine cutting stock problem. We propose mathematical formulations and solution methods to deal with such problems. Firstly, we deal with the two-stage two-dimensional guillotine cutting stock problem in which items are identical, bins are different in size and the objective is to determine the optimal size of the identical items. Two solution procedures are presented to solve the case in which the orientation of the items is fixed and the case in which orthogonal rotation of items is allowed. The two procedures deal with the problem iteratively solving a knapsack problem for each possible item size and returning the best solution found. Numerical experiments are conducted on two- hundred randomly generated instances to evaluate the scalability of the approaches. Lastly, we deal with the k-stage two-dimensional guillotine cutting stock problem in which setup cost associated with stages of cut are considered relevant. A mathematical programming formulation with $O(n^2pk)$ variables and $O(npk)$ constraints is present, in which n, p and k are the number of items, bins and stages, respectively. Numerical experiments are conducted on twenty small-scale randomly generated instances to evaluate the quality of the approach.

Keywords: Cutting and packing problems; identical items; setup; stage cutting; mathematical programming.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

SUMMARY

1 INTRODUCTION

Cutting and Packing Problems (CPP) are a class of combinatorial problems widely studied in the last few decades. Several variants are reported in the revised literature, however, taking into consideration some criteria and constraints, there are several practical characteristics of the problem that have not been addressed.

Two-dimensional cutting problems have the objective of obtaining a set of small objects (items) from a set of large objects (bins). In some practical cases, such as for granite cutting, an additional limitation is imposed regarding the assortment of items known as guillotine cutting. A guillotine cut is defined as an orthogonal edge-to-edge cut applied to an object resulting in two new objects that can be a desired item or a smaller bin to be cut later. A cut stage is a set of parallel guillotine cuts performed on a bin or on the resulting sub-bins from previous stages.

In this work, we present two variants of the two-dimensional guillotine cutting stock problem not addressed in the revised literature so far. Firstly, we deal with the problem of finding the optimal size of the identical items to be obtained from a set of bins of different sizes. Moreover, we deal with a second problem of evaluating the trade-off between material waste associated with the decision regarding cut patterns and setup cost associated with the complexity of cutting patterns used. These problems have a practical relevance as they were identified during technical visits to a local granite industry.

During the visits, several difficulties were highlighted by the budget analyst regarding the cutting planning and customer requirements. According to the analyst, stock pieces are internally produced by cutting granite blocks into rectangular plates and, due to deviation between the dimensions of the blocks, the plates produced are different in size. This heterogeneity of bins is a huge hindrance to the cutting planning considering it is done with the use of spreadsheets.

The company participates in biddings for the supply of materials for the construction of large ventures such as airports and shopping centers, hence the assortment by spreadsheet can take from hours to a couple of days, making the whole process time-consuming and costly.

Another difficulty highlighted is associated with customers requirements related with quotation. Customers may give a list of item types with their respective quantities and request a budget. The quotation is based on the average waste; hence, the final price depends on the cut plane, which depends on the dimensions of the items. Thus, if the budget does not meet the customer expectations, the company may suggest a single standard size for the items that result in the possible lowest budget. For this case, to guarantee a low production cost, it is preferable that the cuts are performed in a maximum of two stages. The difficulty of the process of finding the optimal item size is increased by the heterogeneity of bins and the large number of item size possibilities.

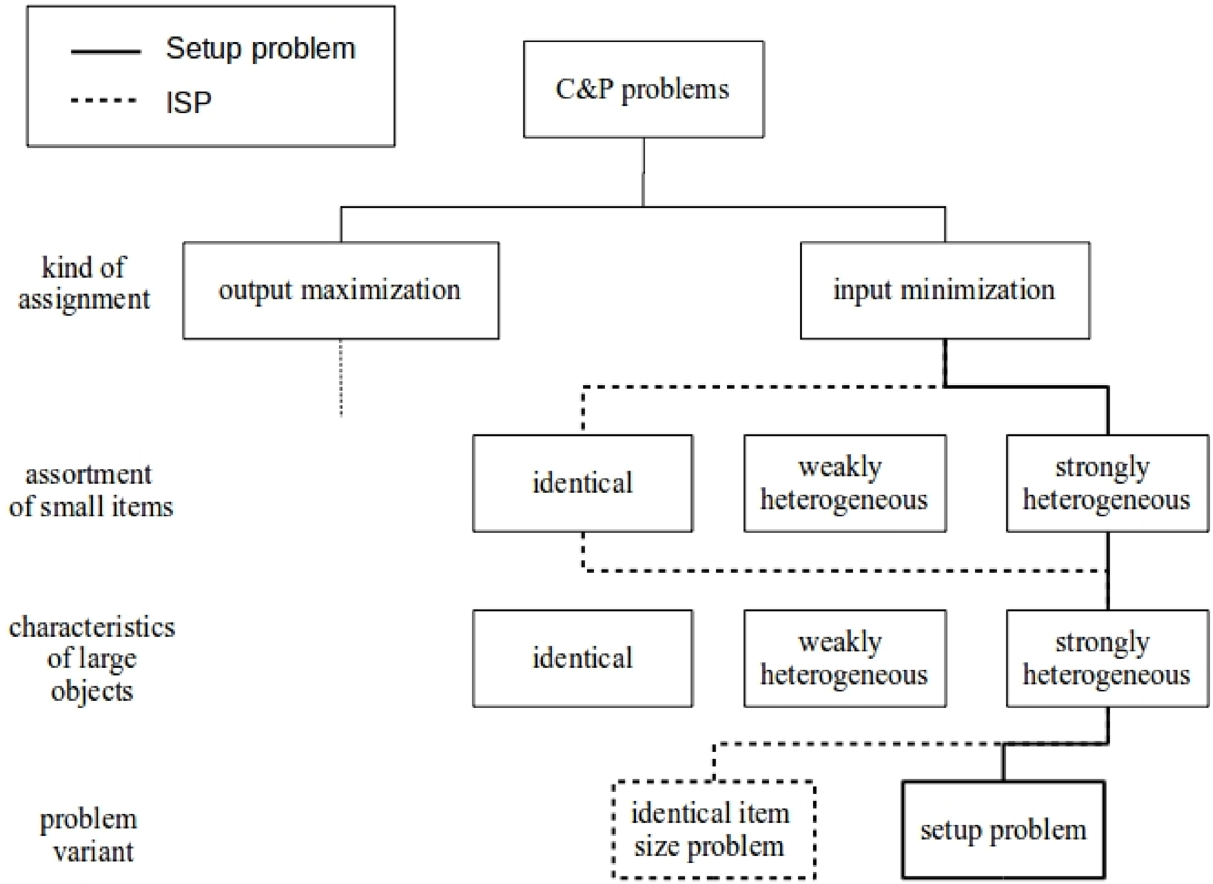
Furthermore, the analyst reported another issue associated with production cost concerning the cut patterns. According to him, depending on the material cost, one can sacrifice the optimization of the plate area usage if the associated cutting plan is operationally easier to be executed. The complexity of a cut plan is associated with the number of cut stages required to obtain the desired items.

Therefore, we define our first problem as a non-exact two-stage two-dimensional guillotine cutting stock problem in which the objective is to decide the size of the identical items to be obtained from a set of bins of different sizes in order to fulfill a demand at minimum cost. Finally, our second problem is defined as a non-exact *k*-stage two-dimensional guillotine cutting stock problem with items and bins of different sizes in which setup cost associated with the use of cut stages is considered.

Figure 1 shows the characteristics of the two problems according to the typology presented by Wäscher *et al.* (2007). Both problems are assigned as input minimization problems, which means that the objective is to minimize cost associated with bin usage. On the item size problem we deal with a set of identical items and, on the setup problem, we deal with a strongly heterogeneous set of items. For both problems, the set of bins is also strongly heterogeneous.

The above-described problems, to the best of our knowledge, have not yet been explored by the revised literature. Therefore, the objective of this work is to present solutions for the above-mentioned problems, opening the field for future solutions ideas and improvements to the solutions presented in this research.

Figure 1 – Contributions of the Thesis



Font: Adapted from Wäscher *et al.* (2007)

2 A TWO-DIMENSIONAL GUILLOTINED CUTTING STOCK PROBLEM CONSIDERING THE OPTIMAL ITEM SIZE DETERMINATION

RESUMO

No presente trabalho é proposta uma nova variante do problema de corte bi-dimensional guilhotinado na qual a decisão está associada a escolha das dimensões h (altura) e w (largura) dos itens idênticos a serem obtidos a partir de um conjunto de placas heterogêneas com o objetivo de atender uma demanda a custo mínimo. Modelos matemáticos e procedimentos são apresentados para solucionar os casos com e sem rotação ortogonal de itens. Os procedimentos lidam com a variante na qual o conjunto de possíveis tipos de itens é finita (discreta), resolvendo iterativamente um modelo matemático para cada tipo de item possível e retornando a melhor solução encontrada. Experimentos computacionais executados em instâncias aleatórias mostram a eficácia da abordagem.

Palavras-chave: Problemas de corte e empacotamento; Itens idênticos; Corte Guilhotinado; Programação matemática, Problema da Mochila.

ABSTRACT

This paper proposes a new variant of the two-dimensional guillotine cutting stock problem in which the decision is associated with the choice of dimensions h (height) and w (width) of the identical items to be cut from a set of heterogeneous stock pieces (bins) in order to fulfill a demand at minimum cost. Mathematical models and solution procedures are presented for solving the cases with and without orthogonal rotational of items. The procedures deal with the variant where the set of possible dimensions for the identical items is finite (discrete) by iteratively solving a mathematical model for each possible item type and returning the best solution found. Computational experiments performed on randomly generated instances show the effectiveness of our approach.

Keywords: Cutting and packing problems; Identical items; Guillotine cut; Mathematical programming; Knapsack problem.

INTRODUCTION

Cutting stock problems appear in several industrial processes where certain materials such as glass, steel, wood or granite are cut to produce smaller items in order to fulfill customer demands. The items may have different shapes such as rectangular, circular or more complex structures. The process of cutting rectangular bins into smaller rectangular items may have technical limitations regarding the equipment used or the material being cut, e.g., some materials such as glass, ceramics and granite may only be cut from edge-to-edge of the material.

The variant proposed in this paper emerged from the granite industry during technical visits to a local manufacture. In this practical case, the bins are different in size and the cut patterns are done manually by the budget analyst. Throughout the visits, the budget analyst highlighted a few difficulties regarding customers requirements, reporting two types of quotation requests. In the first case the customer specifies the sizes and quantities of the items, and, in the second case, the customer specifies the desired total square footage and lets the company decide the size of the identical items.

The budget depends on the average percentage loss, which means that choosing the size of the identical items would allow the possibility of finding a lower budget. In addition, the cuts should preferably be performed in two stages. This premise is based on the need for productivity. Given the variability of sizes of bins and the large number of item size possibilities, this process can become extremely time consuming. Moreover, it is highly unlikely that the analyst will find a good or the best solution as the process is done manually by the use of spreadsheets.

Therefore, we define our problem as a two-stage two-dimensional cutting stock problem with variable bin size in which we decide the dimensions of the identical items. Throughout this paper we will refer to this problem as the Item Size Problem (ISP). The problem under study presents a close relation to the two-dimensional manufacturer's pallet loading problem, more specifically the guillotine restricted case.

Ram (1992) presents a survey on the pallet loading problem. The survey includes considerations in pallet packing, models and solutions methods. Morabito and Morales (1998), Morabito and Farago (2002), Birgin *et al.* (2005), Pureza and Morabito (2006) and Ribeiro and Lorena (2007) have addressed the manufacturer's pallet loading problem.

Wäscher *et al.* (2007) provide a typology of cutting and packing problems improved from Dyckhoff (1990). According to the authors, the identical item packing problem is defined as an input maximization problem, in which a single pallet is loaded with the maximum number of identical boxes.

The problem under study has a restriction associated with the type of cut known as guillotine cutting. Many authors have addressed the guillotine cutting stock problem, e.g, Gilmore and Gomory (1965), MacLeod *et al.* (1993), Christofides and Hadjiconstantinou (1995), Vanderbeck (2001), Alvarez-Valdes *et al.* (2002), Song *et al.* (2004), Belov and Scheithauer (2006), Hifi *et al.* (2012), Morabito and Pureza (2010), Furini and Malaguti (2013), Cui *et al.* (2014), Lodi *et al.* (2015), Andrade *et al.* (2016) and Furini *et al.* (2016).

Steudel (1979) presents a heuristic algorithm to generate pallet loading patterns. The problem is defined as a two-dimensional cutting stock problem in which all the small items have the same dimensions and non-guillotine cuts are allowed. Later, Ghandforoush and Daniels (1992) use a rule-based programming language for determining the packing of identical items inside a large objective with the objective of minimizing waste.

Young-Gun and Kang (2001) deal with the two-dimensional pallet problem with the objective of maximizing the number of identical items loaded onto a large rectangle. The authors propose a new heuristic algorithm to find solutions having a 5-block structure when non-guillotine patterns are allowed. Birgin *et al.* (2005) provide notes on an *L*-approach for solving the manufacturer's pallet loading problem previously studied by Lins *et al.* (2003). Later, Birgin *et al.* (2010) combine an improved version of the 5-block heuristic with an *L*-approach for packing rectangles into larger rectangles and *L*-shaped pieces.

Alvarez-Valdés *et al.* (2005) take the advantage of the relation between the pallet loading problem and the maximum independent set problem to propose a branch-and-cut algorithm for the first problem. Later, Pureza and Morabito (2006) perform experiments with a tabu search algorithm for the non-guillotine manufacturer's pallet loading problem. Birgin and Lobato (2010) propose a mixed integer continuous nonlinear model for the orthogonal packing of identical rectangles within isotropic convex regions, i.e., non-rectangle large objects.

Although the ISP has a strong relationship with the guillotine manufacturer's pallet loading problem, the case in which we decide the size of the identical items, to the best of our knowledge, has not been addressed by the revised literature. Therefore, the objective of this work is to present mathematical formulations and solution methods for the above-mentioned problem. The remaining parts of this paper are structured as follow: (i) in the second section, mathematical formulations are presented; (ii) in the third section, the proposed algorithms and solution methods are exposed; (iii) in the fourth section, the mathematical models and solution methods are numerically tested and the results are discussed; (iv) at the end, the paper presents some conclusions and recommendations for future research.

MATHEMATICAL FORMULATIONS

Non-rotational ISP formulation

The following model, which we will refer to as M_1, concerns the non-rotational ISP. In this model, we consider the set of possible item types to be limited, i.e, the dimensions of the items are discretized. Due to the identical dimensions of items, the assortment of items can easily meet the guillotine restriction by simply packing the items side by side. In what follows, some notation is introduced for the proposed model.

Sets and indexes

$l \in B = \{1, \ldots, p\}$: set of bins;

$i \in I = \{1, \ldots, n\}$ set of item types.

Parameters

H_l: height of bin l;

W_l: width of bin l;

c_l: cost of bin l;

h_i: height of item type i;

w_i: width of item type i;

A: area demand;

$a_{il} = h_i w_i \left\lfloor \frac{W_l}{w_i} \right\rfloor \left\lfloor \frac{H_l}{h_i} \right\rfloor$: useful area of bin l with respect to item type i.

Decision variables

u_{il}: takes the value of 1 if items type i are placed in bin l; 0, otherwise;

y_i: takes the value of 1 if item type i is chosen; 0, otherwise.

Objective function

$$\text{Minimize} \sum_{l \in B} \sum_{i \in I} c_l u_{il} \tag{2.1}$$

Constraints

$$u_{il} \leqslant y_i \quad \forall i \in I, l \in B \tag{2.2}$$

$$\sum_{l \in B} \sum_{i \in I} a_{il} u_{il} \geqslant A \tag{2.3}$$

$$\sum_{i \in I} y_i = 1 \tag{2.4}$$

$$y_i,\, u_{il} \in \{0,1\} \quad \forall i \in I, l \in B \tag{2.5}$$

The objective (2.1) is to minimize total cost. Constraint set (2.2) ensures that items of type i cannot be packed in any bin if the same item is not chosen. Constraint (2.3) ensures demand satisfaction and constraint (2.4) forces the choice of one single item type. Constraint set (2.5) defines the domain of the decision variables. The model has $np+n$ binary variables and $np+2$ constraints.

Rotational ISP formulation

For the second model, the possibility of orthogonal rotation of items is taken into consideration. This consideration allows finding equal or better solutions to those found with fixed item orientation.

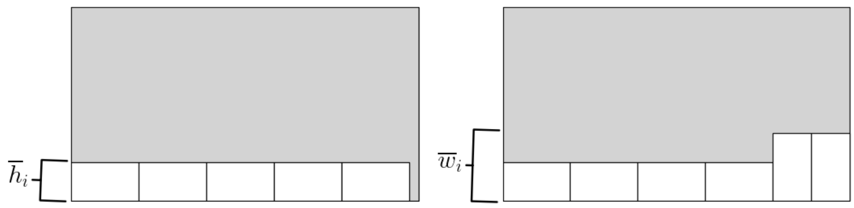
In this case, a pattern based model M_2 is proposed as follows. Let $\overline{h}_i$ and $\overline{w}_i$ be the height and width of item type i such that $\overline{h}_i = \min\{h_i, w_i\}(i \in I)$ and $\overline{w}_i = \max\{h_i, w_i\}(i \in I)$. Let b_{ijl} be the number of items of type i that fits horizontally in bin l when a horizontal unidimensional pattern j is used. These horizontal patterns will be refereed to as strip patterns.

Proposition 2.1 *The maximum number of possible strip patterns for item type i in bin l is equal to* $\left\lfloor \frac{W_l}{\overline{w}_i} \right\rfloor + 1$.

Proof 2.1 *Let the first strip pattern associated with item type i and bin l consist of only non-rotated items. This pattern contains a total of* $k = \left\lfloor \frac{W_l}{\overline{w}_i} \right\rfloor$ *non-rotated items of type i. The following possible strip patterns have* $k-1, k-2, \ldots, 0$ *non-rotated items of type i and* $\left\lfloor \frac{W_l - \overline{w}_i(k-1)}{\overline{h}_i} \right\rfloor, \left\lfloor \frac{W_l - \overline{w}_i(k-2)}{\overline{h}_i} \right\rfloor, \ldots, \left\lfloor \frac{W_l}{\overline{h}_i} \right\rfloor$ *rotated items of type i, respectively, resulting in a total of* $k+1$ *possible strip patterns, i.e.,* $\left\lfloor \frac{W_l}{\overline{w}_i} \right\rfloor + 1$.

The first strip pattern with item type i contains exclusively non-rotated items which means that the height of this strip pattern is equal to $\overline{h}_i$ (Figure 2 - a). The remaining strip patterns contain at least one rotated item, i.e, the height of theses strip patterns is equal to $\overline{w}_i$, since $\overline{h}_i$ is less than or equal to $\overline{w}_i$ for all $i \in I$ (Figure 2 - b).

Figure 2 – Strip pattern example



(a) first strip pattern (b) second strip pattern

Each strip pattern j related to item type i takes a vertical space that can be either $\overline{h}_i$ or $\overline{w}_i$, depending on the presence or absence of a rotated item in the correspondent strip pattern. The cases with presence of rotated items may lead to symmetrical solutions.

Theorem 2.1 *The minimum number of strip patterns associated with item type i that guarantees the maximum area usage of bin l is equal to two.*

Proof 2.2 *The maximum number of items of type i in a strip pattern containing at least one rotated item is equal to* $\left\lfloor \frac{W_l}{\overline{h}_i} \right\rfloor$*, i.e., the one which contains exclusively rotated items. Hence, any other strip pattern containing both rotated and non-rotated items of type i has a number of items that is less than or equal to* $\left\lfloor \frac{W_l}{\overline{h}_i} \right\rfloor$*.The result is two pattern of interest. One containing exclusively non-rotated items with height equal to* $\overline{h}_i$ *and another containing exclusively rotated items with height equal to* $\overline{w}_i$*.*

In order to guarantee all possible solutions, the rotation of bins must also be considered. For this, let B^* be the set of bins including the rotated bins such that $H_l = W_{l+p}$ and

$W_l = H_{l+p}$ for all $l \in B$. This means that set B^* has two times more elements than set B, i.e., $|B^*| = 2|B| = 2p$.

Let x_{ijl} be an integer variable associated with the number of strip patterns j of item type i packed vertically in bin l. Let u_l be a binary variable associated with the use of bin l and let y_i be a binary variable associated with the choice of item type i as the identical item. Let $m_{il} = \left\lfloor \frac{H_l}{\overline{h}_i} \right\rfloor$ be an upper bound for x_{ijl} and let $a_{ijl} = \overline{h}_i \overline{w}_i b_{ijl}$ be the area associated with strip pattern j of item type i with respect to bin l, i.e., $a_{i,1,l} = \overline{h}_i \overline{w}_i \left\lfloor \frac{W_l}{\overline{w}_i} \right\rfloor$ and $a_{i,2,l} = \overline{h}_i \overline{w}_i \left\lfloor \frac{W_l}{\overline{h}_i} \right\rfloor$

Objective function

$$\text{Minimize} \sum_{l \in B^*} c_l u_l \tag{2.6}$$

Constraints

$$\overline{h}_i x_{i1l} + \overline{w}_i x_{i2l} \leqslant H_l u_l \quad \forall l \in B^*, i \in I \tag{2.7}$$

$$x_{ijl} \leqslant m_{il} y_i \quad \forall l \in B^*, i \in I, j = \{1,2\} \tag{2.8}$$

$$\sum_{l \in B^*} \sum_{i \in} \sum_{j=1}^{2} a_{ijl} x_{ijl} \geqslant A \tag{2.9}$$

$$\sum_{i \in I} y_i = 1 \tag{2.10}$$

$$u_l + u_{l+p} \leq 1 \quad \forall l \in B \tag{2.11}$$

$$x_{ijl} \in \mathbb{N}^+ \quad \forall l \in B^*, i \in I, j = \{1,2\} \tag{2.12}$$

$$y_i \in \{0,1\} \quad \forall i \in I \tag{2.13}$$

$$u_l \in \{0,1\} \quad \forall l \in B^* \tag{2.14}$$

The objective (2.6) is to minimize total cost. Constraint set (2.7) limits the sum of heights of the strip patterns packed in bin l to the height of the same bin. Constraint set (2.8) bounds the number of strip patterns of item type i to the choice of the same item as the identical item. Constraint (2.9) guarantees demand satisfaction and constraint (2.10) forces the choice of one single item type. Constraint set (2.11) limits the choice of a non rotated and a rotated bin to one. Constraint sets (2.12), (2.13) and (2.14) define the domain of the decision variables. The model has, at the worse case scenario, $6np+p+2$ constraints, $4np$ integer variables and $n+2p$ binary variables.

ALGORITHMS AND SOLUTION METHODS

In this section, we present two solution procedures for solving the identical item size problem. The procedures deal with the non-rotational case in which the orientation of the identical items is fixed and the rotational case in which orthogonal rotation of items is allowed.

Non-rotational ISP solution method

In models M_1 and M_2, the demand A works as a lower bound, i.e., one can produce more than required, making the two formulations have too many symmetrical solutions. Therefore, the two problems can be approached by fixing the dimensions of the identical items and calculating the maximum area usage for each bin, resulting in a knapsack subproblem, which can be solved for each possible item type. The knapsack subproblem is described as follows.

Sets and indexes

$l \in B = \{1, \ldots, p\}$: set of bins;

$i \in I = \{1, \ldots, n\}$: set of item types;

Parameters

c_l: cost of bin l;

a_l: maximum area usage for bin l;

A: area demand.

Decision variables

u_l: 1, if bin l is used, 0, otherwise.

Objective function

$$\text{Minimize } \sum_{l \in B} c_l u_l \tag{2.15}$$

Constraints

$$\sum_{l \in B} a_l u_l \geqslant A \tag{2.16}$$

$$u_l \in \{0,1\} \quad \forall l \in B \tag{2.17}$$

Taking into consideration the non-rotational case, a_l can be easily enumerated for a fixed item height h and width w by multiplying the amount $\left\lfloor \frac{H_l}{h} \right\rfloor \cdot \left\lfloor \frac{W_l}{w} \right\rfloor$ by the area of the item $h \cdot w$. Thereafter, the non-rotational case can be dealt with, by solving the knapsack subproblem for each possible item type and returning the result with the lowest cost. This knapsack procedure, which we shall refer to as K_1, is detailed in Algorithm 1.

Algorithm 1: Knapsack procedure 1

Data: $H \rightarrow$ array[p], $W \rightarrow$ array[p], $h \rightarrow$ array[n], $w \rightarrow$ array[n], $c \rightarrow$ array[p], $A \rightarrow$ float

Result: bestCost $\rightarrow$ float

bestCost $\leftarrow \infty$;
for $i = 1$ ***to*** n **do**
    for $l = 1$ ***to*** p **do**
        $a_l \leftarrow \left\lfloor \frac{H_l}{h_i} \right\rfloor \cdot \left\lfloor \frac{W_l}{w_i} \right\rfloor \cdot h_i \cdot w_i$;
    end
    cost $\leftarrow$ knapsackSubProblem(a,c,A);
    if *cost* $<$ *bestCost* **then**
        bestCost $\leftarrow$ cost;
    end
end

Rotational ISP solution method

To deal with orthogonal rotation of items, one must first enumerate the parameter a_l by solving a maximization problem. Let $\overline{h} = min\{h,w\}$ and $\overline{w} = max\{h,w\}$ be the height and width of a predetermined item type. Let b_j be the number of items associated with the strip pattern j.

Let H and W be the height and width of a predetermined bin, respectively. Let x_j be an integer variable associated with the number of strip patterns j used. Then the following maximization problem can be solved to find the maximum area usage of a predetermined item type in a predetermined bin. Additionally, $b_1 = \left\lfloor \frac{W}{\overline{w}_i} \right\rfloor$ and $b_2 = \left\lfloor \frac{W}{\overline{h}_i} \right\rfloor$ be the number of items associated with non-rotated and rotated strip patterns, respectively (Proposition 2.1).

Objective function

$$\text{Maximize} \sum_{j=1}^{2} b_j x_j \tag{2.18}$$

Constraints

$$\overline{h} \cdot x_1 + \overline{w} \cdot x_2 \leqslant H \tag{2.19}$$

$$x_j \in \mathbb{N}^+ \quad \forall j = \{1, \ldots, 2\} \tag{2.20}$$

Objective function (2.18) maximizes the number of items packed. Constraint (2.19) limits the sum of heights of the strip patterns used to the height of the bin and constraint set (2.20) defines the domain of the decision variables.

For the rotational case, the knapsack subproblem (2.15)-(2.17) must be adjusted to include the set of rotated bins. The adjusted knapsack subproblem is defined below.

Objective function

$$\text{Minimize} \sum_{l \in B} c_l u_l \tag{2.21}$$

Constraints

$$\sum_{l \in B^*} a_l u_l \geqslant A \tag{2.22}$$

$$u_l + u_{l+p} \leq 1 \quad \forall l \in B \tag{2.23}$$

$$u_l \in \{0, 1\} \quad \forall l \in B^* \tag{2.24}$$

Therefore, the knapsack procedure for solving the rotational problem (K_2) is detailed in Algorithm 2.

Algorithm 2: Solution procedure K_2

Data: $H \rightarrow$ array[2p], $W \rightarrow$ array[2p], $h \rightarrow$ array[n], $w \rightarrow$ array[n], $c \rightarrow$ array[2p], $A \rightarrow$ float

Result: bestCost $\rightarrow$ float

bestCost $\leftarrow \infty$;
for $i = 1$ ***to*** n **do**
  $\overline{h} \leftarrow min(h_i, w_i)$;
  $\overline{w} \leftarrow max(h_i, w_i)$;
  for $l = 1$ ***to*** $2p$ **do**
    $b_1 \leftarrow \left\lfloor \frac{W_l}{\overline{w_i}} \right\rfloor$;
    $b_2 \leftarrow \left\lfloor \frac{W_l}{\overline{h_i}} \right\rfloor$;
    $a_l \leftarrow$ maximizationSubProblem($b, \overline{h}, \overline{w}, H_l$);
  end
  cost $\leftarrow$ adjustedKnapsackSubProblem(a, c);
  if *cost* $<$ *bestCost* **then**
    bestCost $\leftarrow$ cost;
  end
end

Additionally, one can reduce the size of set I (set of possible items) by removing dominated item types as detailed in Algorithm 3. The result is a new set I^* where $|I^*| \leqslant |I|$.

Theorem 2.2 *If $a_l^{(k)}$ is the maximum area usage of bin l with respect to item type k and $a_l^{(k)} \leqslant a_l^{(i)}$ for all $l \in B$, then item type k is dominated and can be removed from set I of possible items.*

Proof 2.3 *If $a_l^{(k)} \leqslant a_l^{(i)}$ for all $l \in B$ then $\sum_{l \in B} a_l^{(k)} u_l \leqslant \sum_{l \in B} a_l^{(i)} u_l$. Since $\sum_{l \in B} a_l^{(k)} u_l \geqslant A$ and $\sum_{l \in B} a_l^{(i)} u_l \geqslant A$, item type i generates a solution that is better than or equal to that generated with item type k, i.e., i is dominant with respect to k.*

Algorithm 3: Dominated items elimination procedure

Data: $a \rightarrow$ array[n, p], $I \rightarrow$ array[n]

Result: $I^* \rightarrow$ array[$n^* \leqslant n$]

```
I* ← I;
for j = 1 to n do
    for i = 1 to n do
        win ← False;
        if i ≠ j then
            for l = 1 to p do
                if a_l^(j) > a_l^(i) then
                    win ← True;
                    breakLoop;
                end
            end
            if win = False then
                remove j from I*;
                breakLoop;
            end
        end
    end
end
```

COMPUTATIONAL RESULTS

In this work, we present some numerical experiments on the two mathematical models and on the two solution procedures with the use of forty randomly generated instances. In order to generate the instances, a set of four bin quantities $10, 15, 20$ and 25 and a set of ten item type quantities 50, 100, 150, 200, 250, 300, 350, 400, 450 and 500 are defined, resulting in a total of 40 instances. The dimensions of the bins and item types are set as a discrete uniform distribution with minimum and maximum of 100 and 200 for the bins and with minimum and maximum of 10 and 60 for the item types. (https://www.researchgate.net/publication/322937332_random_instances).

The models and solution procedures were implemented using programing language

Python version 3.5 with Pyomo mathematical modeling package (http://www.pyomo.org/) and IBM ILOG CPLEX 12.6.0. The experiments were conducted on a machine with Intel® Core ™ i5-5200U CPU 2.20GHz x 4 processor, 4GB of RAM, running Ubuntu 16.04 LTS.

For the computational tests we set $c_l = H_l W_l$, which means that cost is proportional to the area of the bins. Additionally, a time limit of 3600 seconds is imposed to CPLEX for solving models M_1 and M_2, which will return the relative gap for the cases where time limit is reached. Tables 1 and 2 present the computational results for the cases with and without rotation of items, respectively, including the results for the mathematical models and the solution procedures.

In Tables 1 and 2, the first column indicates the instance which is being evaluated followed by the number of bins and items associated with the respective instance. The forth column shows the number of items after the dominated item elimination procedure. The next four columns are associated with the global optimal solution, enumeration time (including the dominated item elimination procedure), processing time and total time for the solution procedure, respectively. The three last columns show the global optimal solution, relative gap and total computational time for the mathematical model, respectively. Additionally, Figures 5 and 6 display the comparison between the total computational time of the solution procedure and the mathematical model for the non-rotational and rotational cases, respectively.

The enumeration time of procedure K_1 is, on average, only 1.46% of total computing time, in contrast to 98.61% for K_2, as observed in Tables 1 and 2. This large gap between the two procedures is related to the use of mathematical programming for the enumeration procedure by K_2 combined with the larger number of dominant patterns. The result is an average computational time ratio of approximately 212 between K_2 and K_1, i.e., solution procedure K_2 takes, on average, two-hundred and twelve times more computational effort in comparison to K_1. Additionally, on average, the size of the item type set with dominated item elimination $|I^*|$ is 57.91% smaller than the size of set $|I|$ with respect to K_1, in contrast to 2.65% with respect to K_2.

By analyzing Figure 5 (and Table 1), we observe that, for all instances evaluated, model M_1 is superior to procedure K_1 in terms of computational effort. An opposite behaver is observed in the rotational case (Figure 6) in which model M_2 is far inferior to procedure K_2 in terms of computational effort. Model M_2 was able to find the global optimal solution for the first six instances and instance eleven, however, for the remaining instances, the model reached

Table 1 – Computational results (non-rotational)

			Procedure K_1					Model M_1			
inst.	$\|B\|$	$\|I\|$	$\|I^*\|$	sol.	enu. time	proc. time	total time	sol.	gap	lin. relax. gap	total time
1	10	50	23	43467	0.0	0.5	0.5	43467	0.0	0.0	0.1
2	10	100	31	46936	0.0	0.6	0.6	46936	0.0	0.0	0.2
3	10	150	40	52795	0.0	1.2	1.2	52795	0.0	0.0	0.2
4	10	200	53	44456	0.0	1.2	1.2	44456	0.0	0.0	0.2
5	10	250	68	60815	0.0	1.4	1.4	60815	0.0	0.0	0.4
6	10	300	86	40302	0.0	1.8	1.8	40302	0.0	0.0	0.4
7	10	350	55	49103	0.0	1.2	1.2	49103	0.0	0.0	0.5
8	10	400	96	56105	0.1	2.0	2.1	56105	0.0	0.0	0.5
9	10	450	94	36741	0.1	2.0	2.1	36741	0.0	0.0	0.4
10	10	500	93	47688	0.1	1.9	2.0	47688	0.0	0.0	0.6
11	15	50	37	74475	0.0	0.8	0.8	74475	0.0	0.0	0.1
12	15	100	67	75446	0.0	1.5	1.5	75446	0.0	0.0	0.1
13	15	150	62	64936	0.0	1.4	1.4	64936	0.0	0.0	0.4
14	15	200	75	68602	0.0	1.7	1.7	68602	0.0	0.0	0.4
15	15	250	74	74652	0.0	1.8	1.8	74652	0.0	0.0	0.3
16	15	300	105	63485	0.1	2.4	2.5	63485	0.0	0.0	0.3
17	15	350	109	67864	0.1	2.4	2.5	67864	0.0	0.0	0.5
18	15	400	99	71115	0.1	2.3	2.4	71115	0.0	0.0	0.5
19	15	450	150	72750	0.1	3.6	3.7	72750	0.0	0.0	1.1
20	15	500	143	69564	0.1	3.5	3.6	69564	0.0	0.0	1.3
21	20	50	42	94016	0.0	1.1	1.1	94016	0.0	0.0	0.1
22	20	100	62	87756	0.0	1.6	1.6	87756	0.0	0.0	0.2
23	20	150	69	91596	0.0	1.9	1.9	91596	0.0	0.0	0.3
24	20	200	117	97001	0.0	3.3	3.3	97001	0.0	0.0	0.4
25	20	250	126	93932	0.1	3.4	3.5	93932	0.0	0.0	0.3
26	20	300	131	91309	0.1	4.5	4.6	91309	0.0	0.0	0.5
27	20	350	140	89152	0.1	5.2	5.3	89152	0.0	0.0	0.8
28	20	400	150	91398	0.1	6.4	6.5	91398	0.0	0.0	0.6
29	20	450	142	88850	0.1	6.5	6.6	88850	0.0	0.0	0.6
30	20	500	143	97658	0.1	7.8	7.9	97658	0.0	0.0	1.2
31	25	50	38	119468	0.0	2.6	2.6	119468	0.0	0.0	0.2
32	25	100	65	114516	0.0	6.2	6.2	114516	0.0	0.0	0.3
33	25	150	82	120242	0.0	6.7	6.7	120242	0.0	0.0	0.3
34	25	200	113	111216	0.0	9.4	9.4	111216	0.0	0.0	0.5
35	25	250	141	114969	0.1	11.1	11.2	114969	0.0	0.0	0.4
36	25	300	149	120737	0.1	12.2	12.3	120737	0.0	0.0	0.8
37	25	350	150	113936	0.1	12.8	12.9	113936	0.0	0.0	0.9
38	25	400	196	106664	0.1	16.0	16.1	106664	0.0	0.0	0.9
39	25	450	181	115959	0.1	15.8	15.9	115959	0.0	0.0	1.1
40	25	500	218	107034	0.2	16.9	17.1	107034	0.0	0.0	1.2

the time limit of 3600 seconds with positive linear relaxation gaps from 12.1% up to 99.4%. From this point we will focus on the analysis of the results for model M_1 and procedure K_2 in detriment to their superior performances in comparison to K_1 and M_2, respectively.

As expected, the global optimal solution values for the rotational case are less than or equal to the values for the non-rotational case for all the instances evaluated. Figures 3 and 4

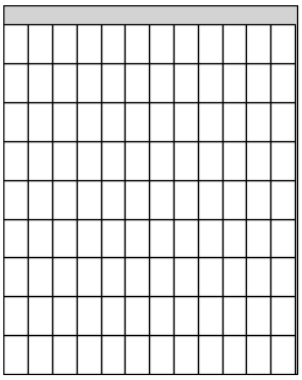
Table 2 – Computational results (rotational)

inst.	$\|B\|$	$\|I\|$	Procedure K_2: $\|I^*\|$	sol.	enu. time	proc. time	total time	Model M_2: sol.	gap	lin. relax. gap	total time
1	10	50	43	41764	65.3	0.9	66.2	41764	0.0	0.0	42.1
2	10	100	99	45570	140.3	2.1	142.4	45570	0.0	0.0	180.0
3	10	150	132	50863	225.0	3.1	228.1	50863	0.0	0.0	293.1
4	10	200	189	43945	311.2	4.7	315.9	43945	0.0	0.0	969.3
5	10	250	219	58590	341.6	4.8	346.4	58590	0.0	0.0	1552.3
6	10	300	297	38918	433.2	6.5	439.7	38918	0.0	0.0	2533.3
7	10	350	349	48233	509.2	7.6	516.8	48233	0.0	57.9	3600.2
8	10	400	387	54972	581.3	8.6	589.9	54972	0.0	58.8	3600.2
9	10	450	446	36375	652.0	9.9	661.9	36375	0.0	63.0	3600.2
10	10	500	454	47280	708.2	10.0	718.2	47688	0.9	69.3	3600.6
11	15	50	49	72504	110.3	1.2	111.5	72504	0.0	0.0	896.8
12	15	100	100	74672	221.6	2.5	224.1	75159	0.7	12.1	3600.2
13	15	150	141	62795	322.6	3.5	326.1	63685	1.4	64.2	3603.3
14	15	200	200	66772	443.3	4.8	448.1	67536	1.1	73.2	3600.1
15	15	250	237	73723	536.6	6.0	542.6	74523	1.1	80.0	3600.2
16	15	300	300	62817	661.7	7.4	669.1	63220	0.6	87.7	3600.2
17	15	350	345	66737	780.6	8.7	789.3	67164	0.6	91.5	3600.2
18	15	400	398	70076	891.1	10.0	901.1	71535	2.1	91.4	3600.2
19	15	450	446	70299	1041.2	12.0	1053.2	70776	0.7	99.4	3600.3
20	15	500	492	68136	1156.0	13.6	1169.6	70288	3.2	91.2	3600.3
21	20	50	50	92519	153.5	1.4	154.9	92519	0.0	60.3	3605.3
22	20	100	99	85600	304.8	2.9	307.7	86002	0.5	68.2	3600.2
23	20	150	148	89702	465.7	4.4	470.1	91700	2.2	79.6	3600.1
24	20	200	200	93770	636.8	6.4	643.2	96390	2.8	90.6	3600.2
25	20	250	249	92808	807.8	9.4	817.2	94370	1.7	89.6	3600.4
26	20	300	298	89605	964.0	12.1	976.1	91914	2.6	94.1	3600.2
27	20	350	350	87349	1156.5	15.3	1171.8	89890	2.9	93.0	3600.2
28	20	400	399	89822	1328.3	19.7	1348.0	90641	0.9	91.7	3600.5
29	20	450	449	87908	1527.9	23.8	1551.7	89558	1.9	96.6	3600.3
30	20	500	478	95319	1700.3	34.6	1734.9	97977	2.8	99.0	3600.4
31	25	50	50	115343	216.1	3.8	219.9	115343	0.0	68.7	3600.1
32	25	100	86	111856	398.6	7.8	406.4	114095	2.0	85.2	3600.1
33	25	150	149	117436	642.0	13.4	655.4	117978	0.5	91.2	3600.2
34	25	200	200	109419	851.7	16.6	868.3	111038	1.5	94.4	3600.2
35	25	250	249	113458	1090.2	20.2	1110.4	115954	2.2	93.7	3600.5
36	25	300	297	118730	1304.7	24.9	1329.6	121259	2.1	91.7	3600.4
37	25	350	350	112290	1548.1	29.4	1577.5	114843	2.3	93.0	3600.3
38	25	400	390	103784	1739.8	33.3	1773.1	106916	3.0	93.8	3600.7
39	25	450	447	113366	1961.7	36.9	1998.6	115351	1.8	97.7	3600.4
40	25	500	493	105065	2104.5	15.4	2119.9	108739	3.5	94.4	3600.5

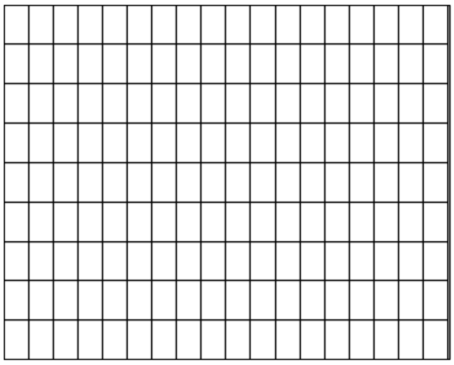
illustrate the solution for instance 4 with respect to model M_1 and procedure K_2, respectively.

Figures 7 and 8 present the correlation between total computational time and the number of items for different numbers of bins. The linear pattern is expected for procedure K_2 as the number of iterations has a linear relation with the size of set I^*. For M_1 (Figure 7), the impact of the number of bins in the computational time is minimum as the angular coefficient of

Figure 3 – Solution for instance 4 - K_1

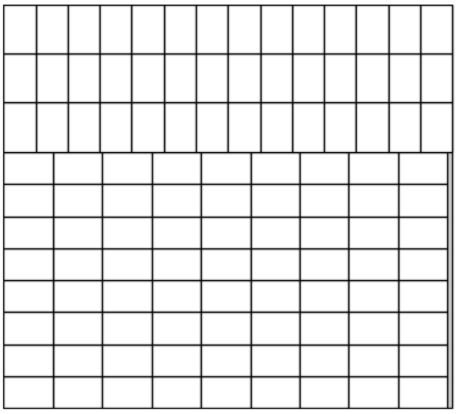


(a) Bin: 121×152 Item: 10×16

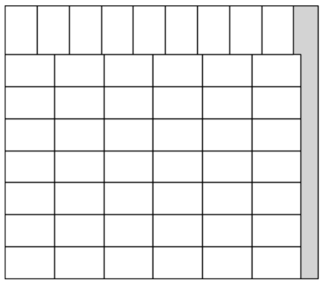


(b) Bin: 181×144 Item: 10×16

Figure 4 – Solution for instance 4 - K_2



(a) Bin: 182×164 Item: 20×13



(b) Bin: 127×111 Item: 20×13

Figure 5 – Computational time (non-rotational)

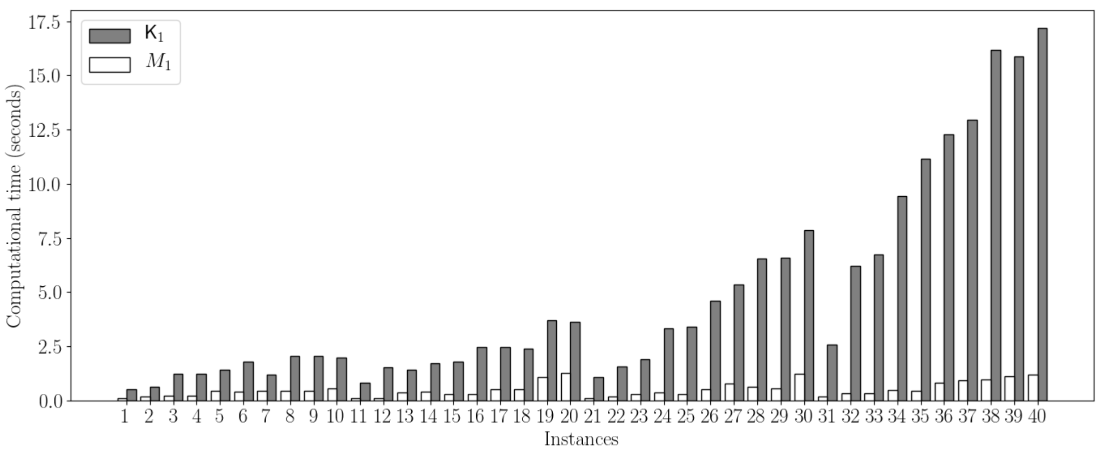


Figure 6 – Computational time (rotational)

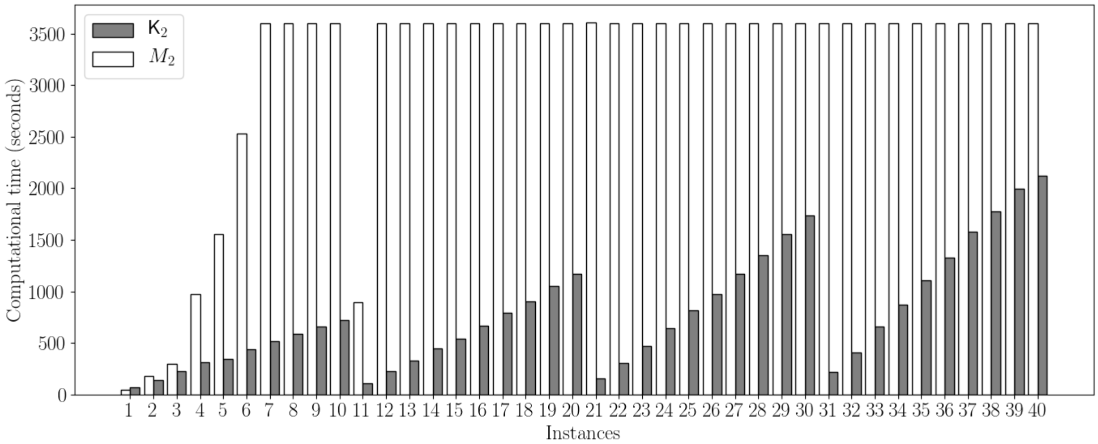
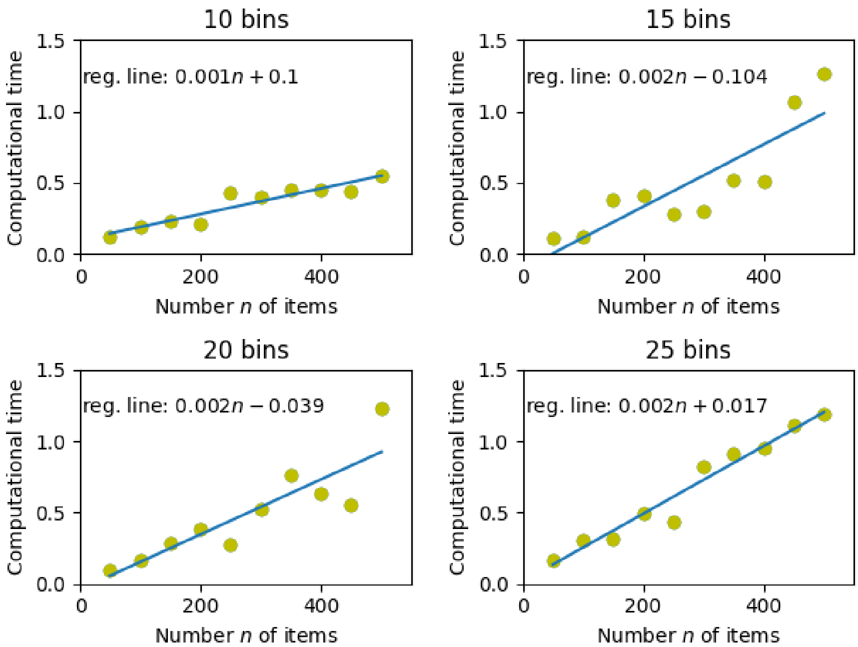
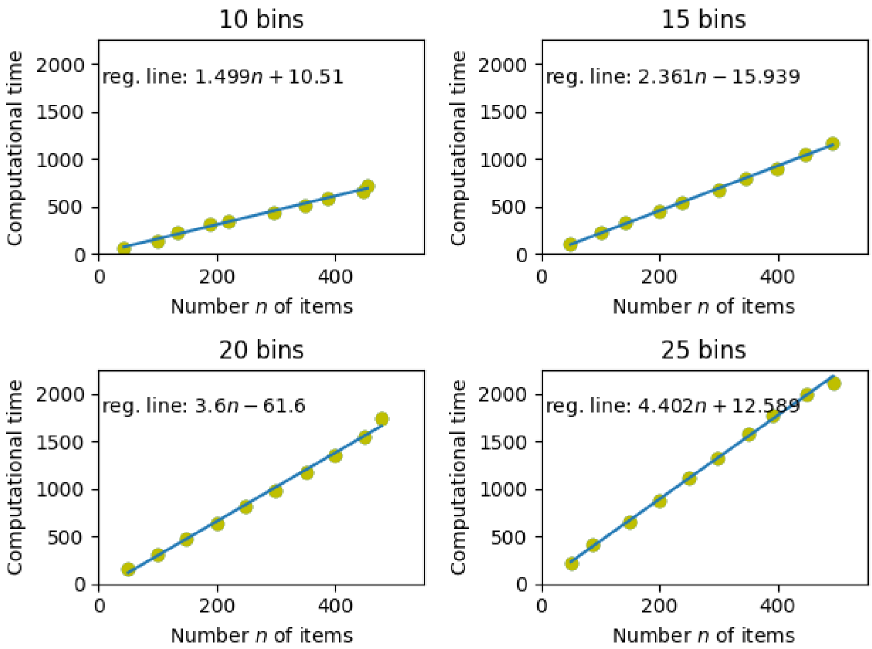


Figure 7 – Computational time x Number of items (M_1)



the regression line had an insignificant change. However, for procedure K$_2$ (Figure 8), the impact is clear. The reason for this behavior relies on the fact that the size of set I^* has a direct impact on the number of times the knapsack subproblem is solved. Therefore, the use of mathematical programming for the enumeration by procedure K$_2$ combined with the larger size of set I^* increases the solution time substantially.

Figure 8 – Computational time x Number of items (K_2)



CONCLUSIONS

In this research, a new variant of the two-stage two-dimensional guillotine cutting stock problem was presented. The variant dealt with the case in which items are identical, bins are different in size and the objective is to find the optimum size of the items. This problem was identified in the granite industry during technical visits to a local manufacturer and, to the best of our knowledge; there is no reference of this problem in the available literature.

Mathematical formulations and solution procedures are presented for the cases with and without orthogonal rotation of items. The proposed approaches were evaluated with the use of forty randomly generated instances. The proposed methodology was able to find the global optimum solution for all used instances with acceptable computational time.

For the case without rotation of items, the mathematical model M_1 was able to outperform the solution procedure K_1 for all instances evaluated. For the case with rotation of item, however, the solution procedure K_2 presented far better results than model M_2. The developed solution procedures dealt with the problem by interactively solving a knapsack subproblem for each possible item size and returning the best solution found. Therefore, the set of possible item sizes is considered finite (or discrete). This assumption was made by virtue of practical operational factors such as cutting precision and standardization of item sizes.

Although our approach has proven to be effective, some improvements can be explored in the future. The first possible improvement is to extend the solution procedure for the case with orthogonal rotation of items (K_2) from a two-stage to a k-stage guillotine cutting stock procedure. A second possible improvement is to consider the possibility of choosing more the one item. Lastly, one could improve the computational efficiency of the solution procedures by parallel programming. This last suggestion could lead procedure K_1 to outperform model M_1.

REFERENCES

ALVAREZ-VALDES, R.; PARAJON, A.; TAMARIT, J. M. A computational study of lp-based heuristic algorithms for two-dimensional guillotine cutting stock problems. **OR Spectrum**, Springer, v. 24, n. 2, p. 179–192, 2002.

ALVAREZ-VALDÉS, R.; PARREÑO, F.; TAMARIT, J. M. A branch-and-cut algorithm for the pallet loading problem. **Computers & Operations Research**, Elsevier, v. 32, n. 11, p. 3007–3029, 2005.

ANDRADE, R.; BIRGIN, E.; MORABITO, R. Two-stage two-dimensional guillotine cutting stock problems with usable leftover. **International Transactions in Operational Research**, Wiley Online Library, v. 23, n. 1-2, p. 121–145, 2016.

BELOV, G.; SCHEITHAUER, G. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. **European journal of operational research**, Elsevier, v. 171, n. 1, p. 85–106, 2006.

BIRGIN, E. G.; LOBATO, R. D. Orthogonal packing of identical rectangles within isotropic convex regions. **Computers & Industrial Engineering**, Elsevier, v. 59, n. 4, p. 595–602, 2010.

BIRGIN, E. G.; LOBATO, R. D.; MORABITO, R. An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. **Journal of the Operational Research Society**, Springer, v. 61, n. 2, p. 306–320, 2010.

BIRGIN, E. G.; MORABITO, R.; NISHIHARA, F. H. A note on an l-approach for solving the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, Springer, v. 56, n. 12, p. 1448–1451, 2005.

BORTFELDT, A.; JUNGMANN, S. A tree search algorithm for solving the multi-dimensional strip packing problem with guillotine cutting constraint. **Annals of Operations Research**, Springer, v. 196, n. 1, p. 53–71, 2012.

CHRISTOFIDES, N.; HADJICONSTANTINOU, E. An exact algorithm for orthogonal 2-d cutting problems using guillotine cuts. **European Journal of Operational Research**, Elsevier, v. 83, n. 1, p. 21–38, 1995.

CINTRA, G.; MIYAZAWA, F. K.; WAKABAYASHI, Y.; XAVIER, E. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. **European Journal of Operational Research**, Elsevier, v. 191, n. 1, p. 61–85, 2008.

CUI, Y.; CUI, Y.-P.; YANG, L. Heuristic for the two-dimensional arbitrary stock-size cutting stock problem. **Computers & Industrial Engineering**, Elsevier, v. 78, p. 195–204, 2014.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, Elsevier, v. 44, n. 2, p. 145–159, 1990.

FURINI, F.; MALAGUTI, E. Models for the two-dimensional two-stage cutting stock problem with multiple stock size. **Computers & Operations Research**, Elsevier, v. 40, n. 8, p. 1953–1962, 2013.

FURINI, F.; MALAGUTI, E.; THOMOPULOS, D. Modeling two-dimensional guillotine cutting problems via integer programming. **INFORMS Journal on Computing**, INFORMS, v. 28, n. 4, p. 736–751, 2016.

GHANDFOROUSH, P.; DANIELS, J. J. A heuristic algorithm for the guillotine constrained cutting stock problem. **ORSA journal on Computing**, INFORMS, v. 4, n. 3, p. 351–356, 1992.

GILMORE, P.; GOMORY, R. E. Multistage cutting stock problems of two and more dimensions. **Operations research**, INFORMS, v. 13, n. 1, p. 94–120, 1965.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. **Operations research**, INFORMS, v. 9, n. 6, p. 849–859, 1961.

HIFI, M. Exact algorithms for the guillotine strip cutting/packing problem. **Computers & Operations Research**, Elsevier, v. 25, n. 11, p. 925–940, 1998.

HIFI, M.; NEGRE, S.; OUAFI, R.; SAADI, T. A parallel algorithm for constrained two-staged two-dimensional cutting problems. **Computers & Industrial Engineering**, Elsevier, v. 62, n. 1, p. 177–189, 2012.

LINS, L.; LINS, S.; MORABITO, R. An l-approach for packing (l, w)-rectangles into rectangular and l-shaped pieces. **Journal of the Operational Research Society**, Springer, v. 54, n. 7, p. 777–789, 2003.

LODI, A.; MONACI, M.; PIETROBUONI, E. Partial enumeration algorithms for two-dimensional bin packing problem with guillotine constraints. **Discrete Applied Mathematics**, Elsevier, 2015.

MACLEOD, B.; MOLL, R.; GIRKAR, M.; HANIFI, N. An algorithm for the 2d guillotine cutting stock problem. **European Journal of Operational Research**, Elsevier, v. 68, n. 3, p. 400–412, 1993.

MORABITO, R.; FARAGO, R. A tight lagrangean relaxation bound for the manufacturer's pallet loading problem. **Stud. Inform. Univ.**, v. 2, n. 1, p. 57–76, 2002.

MORABITO, R.; MORALES, S. A simple and effective recursive procedure for the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, Palgrave Macmillan, v. 49, n. 8, p. 819–828, 1998.

MORABITO, R.; PUREZA, V. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. **Annals of Operations Research**, Springer, v. 179, n. 1, p. 297–315, 2010.

NTENE, N.; VUUREN, J. H. van. A survey and comparison of guillotine heuristics for the 2d oriented offline strip packing problem. **Discrete Optimization**, Elsevier, v. 6, n. 2, p. 174–188, 2009.

PUREZA, V.; MORABITO, R. Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. **Computers & Operations Research**, Elsevier, v. 33, n. 3, p. 804–819, 2006.

RAM, B. The pallet loading problem: A survey. **International Journal of Production Economics**, Elsevier, v. 28, n. 2, p. 217–225, 1992.

RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. **Computers & Operations Research**, Elsevier, v. 34, n. 9, p. 2695–2708, 2007.

SILVA, E.; ALVELOS, F.; CARVALHO, J. V. de. An integer programming model for two-and three-stage two-dimensional cutting stock problems. **European Journal of Operational Research**, Elsevier, v. 205, n. 3, p. 699–708, 2010.

SONG, X.; CHU, C.; NIE, Y. A heuristic dynamic programming algorithm for 2d unconstrained guillotine cutting. **WSEAS Transactions on Mathematics**, v. 3, p. 230–238, 2004.

STEUDEL, H. J. Generating pallet loading patterns: a special case of the two-dimensional cutting stock problem. **Management Science**, INFORMS, v. 25, n. 10, p. 997–1004, 1979.

SULIMAN, S. A sequential heuristic procedure for the two-dimensional cutting-stock problem. **International Journal of Production Economics**, Elsevier, v. 99, n. 1, p. 177–185, 2006.

VANDERBECK, F. A nested decomposition approach to a three-stage, two-dimensional cutting-stock problem. **Management Science**, INFORMS, v. 47, n. 6, p. 864–879, 2001.

WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European journal of operational research**, Elsevier, v. 183, n. 3, p. 1109–1130, 2007.

WEI, L.; TIAN, T.; ZHU, W.; LIM, A. A block-based layer building approach for the 2d guillotine strip packing problem. **European Journal of Operational Research**, Elsevier, v. 239, n. 1, p. 58–69, 2014.

YOUNG-GUN, G.; KANG, M.-K. A fast algorithm for two-dimensional pallet loading problems of large size. **European Journal of Operational Research**, Elsevier, v. 134, n. 1, p. 193–202, 2001.

3 A *K*-STAGE TWO-DIMENSIONAL GUILLOTINE CUTTING STOCK PROBLEM WITH SETUP COSTS

RESUMO

Neste estudo é investigado o problema de corte de k-estágios, no qual é considerado o custo de *setup* associado aos estágios de corte. Esta nova variante aparece na indústria do granito, na qual o *trade-off* entre o custo de desperdício de material e o custo de operação deve ser avaliado em cada licitação. Uma formulação matemática pseudo-polinomial com $O(2^{3n}p)$ variáveis e $O(2^{2n}p)$ restrições é apresentada, on qual n e p são o número de itens e placas, respectivamente. Testes computacionais são elaborados em seis instâncias especiais geradas aleatoriamente.

Palavras-chave: Problemas de corte e empacotamento; Programação matemática

ABSTRACT

In this study we investigate the k-stage two-dimensional guillotine cutting stock problem, in which setup cost associated with cut stages is considered. This new variant appears in the granite industry, in which the trade-off between material waste and operational cost must be evaluated in each bid. A pseudo-polynomial mathematical formulation with $O(2^{3n}p)$ variables and $O(2^{2n}p)$ constraints is presented, in which n and p are the number of items and bins, respectively. Computational tests are performed in six special randomly-generated instances.

Keywords: Cutting and packing problems; mathematical programming.

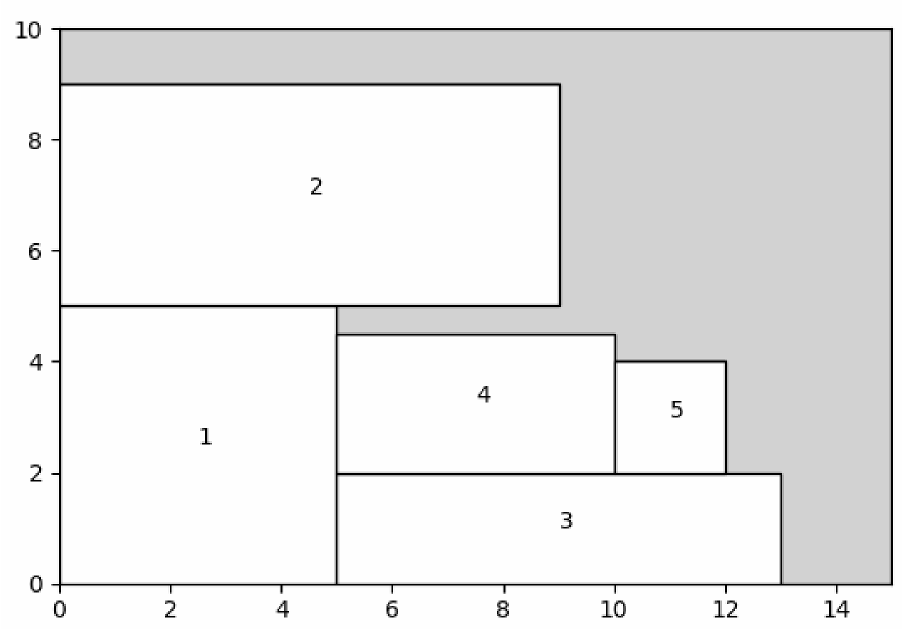
INTRODUCTION

Cutting and packing problems (CPP) are a class of combinatorial problems widely studied in the past fifty years. A special class of CPP is known as two-dimensional guillotine cutting stock problem in which large rectangular objects (bins) are orthogonally cut from edge-to-edge in order to produce small objects (items). A set of parallel edge-to-edge cuts configures a stage of cut and each stage has an orientation that sequentially alternates between horizontal and vertical starting with the chosen orientation for the first stage.

The variant proposed in this paper was identified during technical visits to a local granite mill. Throughout the visits, the budget analyst highlighted two major difficulties regarding cut planning. The first one is associated with the heterogeneity of the stock. According to the analyst, the different sizes of bins makes the planning process more difficult as the number of possible cutting patterns is greater than it would be if the bins were identical. The second difficulty is associated with the trade-off between operational cost and complexity level of a certain cut plan. The operational cost and the complexity are associated with the number of setups required to execute a cut plan.

In this specific practical case, the cutting blade performs edge-to-edge cuts with fixed orientation, which means that, every time a new cutting stage is used, a setup (i.e. bin rotation) is required. The problem relies on the fact that setup is significant to the total cost. In Figure 9, e.g., item 5 have a higher setup cost in compare to item 2 as item 5 takes three stages of cutting to be extracted from the bin in contrast to one stage for item 2, as illustrated in Figure 10.

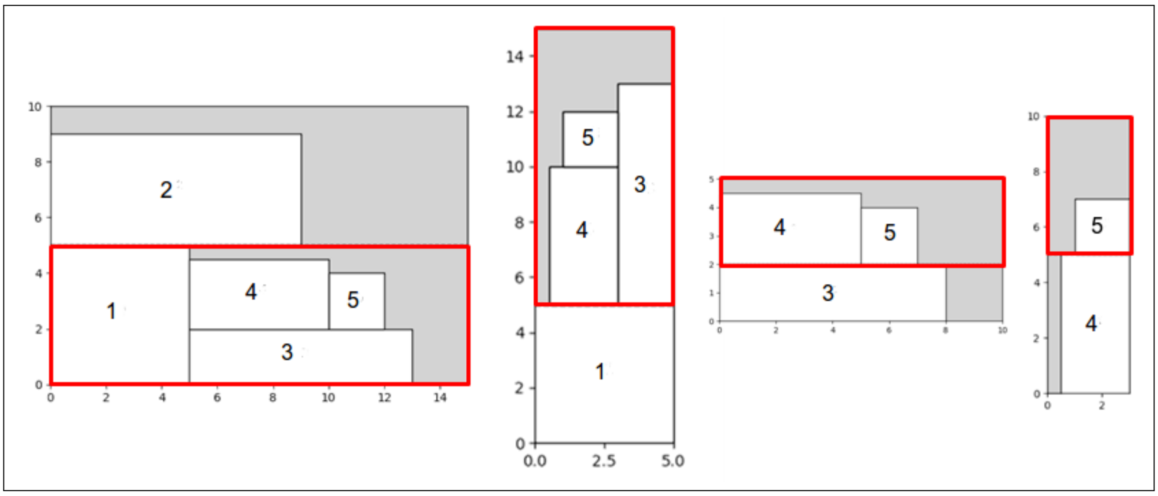
Figure 9 – Four-stage cutting pattern



According to the typology of Wäscher *et al.* (2007) the problem addressed in this section is a two-dimensional rectangular Multiple Stock Size Cutting Stock Problem (MSSCSP) with additional constraints related with the types of cuts allowed and additional cost associated with setup. In this particular case, the additional constraints regarding the types of cut allowed is related to orthogonal edge-to-edge cuts, i.e., guillotine cuts.

Many authors have addressed the guillotine cutting stock problem, e.g, Gilmore and Gomory (1965), MacLeod *et al.* (1993), Christofides and Hadjiconstantinou (1995), Vanderbeck

Figure 10 – Four-stage cutting procedure



(2001), Alvarez-Valdes *et al.* (2002), Song *et al.* (2004), Belov and Scheithauer (2006), Morabito and Pureza (2010), Furini and Malaguti (2013), Lodi *et al.* (2015), Andrade *et al.* (2016) and Furini *et al.* (2016).

An $O(n^3)$ approximation algorithm for the two-dimensional guillotine cutting stock problem is presented by MacLeod *et al.* (1993). A column generation based approach for a three-stage two-dimensional cutting stock problem is used by Vanderbeck (2001) with additional constraints related to the cutting process. Later, Alvarez-Valdes *et al.* (2002) and Cintra *et al.* (2008) also present a column generation procedure with the use of dynamic programming.

Suliman (2006) presents a three stages sequential heuristic procedure for the two-dimensional cutting-stock problem without guillotine constraint. Belov and Scheithauer (2006) use a branch-and-cut-and-price algorithm for one-dimensional cutting stock problem and two-dimensional two-stage cutting stock problem based on Gilmore and Gomory (1961) and Gilmore and Gomory (1965). Later, Cintra *et al.* (2008) investigate several two-dimensional guillotine cutting stock and strip packing problem variants with orthogonal rotation of items.

The unconstrained variant of the two-dimensional unconstrained guillotine cutting is investigated by Song *et al.* (2004) with the use of dynamic programming. Later, Morabito and Pureza (2010) presents a heuristic approach based on dynamic programming and and/or graph search for the constrained two-dimensional guillotine cutting problem using a state space relaxation of a dynamic programming formulation of the problem and a state space ascent procedure of sub-gradient optimization type. Lodi *et al.* (2015) present a heuristic algorithm based on partial enumeration to solve the two-dimensional bin packing problem in which items must be obtained by a series of guillotine cuts without orthogonal rotation.

Silva *et al.* (2010) propose an integer-programming model for two and three-stage two-dimensional cutting stock problems considering the input minimization case for the exact and non-exact problems. Later, Furini and Malaguti (2013) present three mixed integer problem models for the two-dimensional two-stage cutting stock problem with multiple stock size with a polynomial, pseudo-polynomial and exponential number of variables, respectively. Andrade *et al.* (2016) propose two models for the two-dimensional cutting stock problem with usable leftovers in which the bins present different size.

Hifi (1998) and later Bortfeldt and Jungmann (2012) propose algorithms for the guillotine strip packing in which the first is a tree search algorithm for the strip packing problem in two and three dimensions and the second is a branch-and-bound and dynamic programming procedure for the two dimensional case. Wei *et al.* (2014) approach the two-dimensional guillotine strip packing problem with a block-based layer building in which three known strategies are combined in a single algorithm. For more insight on heuristics for the two-dimensional guillotine strip packing see Ntene and Vuuren (2009).

This paper aims at presenting and solving a new variant of CCP defined as a k-stage two-dimensional guillotine cutting stock problem with setup cost associated with stages. To the best of our knowledge, this work is the first to address the above-mentioned problem. The rest of this paper is structured as follows: in the next section, a mathematical formulation is presented; in the third section, the computational results are shown. Lastly, in the final section, we present some conclusions and suggestions for future research.

MATHEMATICAL FORMULATION

The following model is an extension of the bin packing model M_1 of Andrade *et al.* (2016). The author deals with two-stage bin packing problems in which items open horizontal spaces inside a bin, known as shelves. The height and width of a particular shelf is equal to the height of the item that opened and the difference between the width of the bin and the widths of the item, respectively. Finally, the remaining items can be packed inside the opened shelves. The model presented in this paper extends this idea for approaching a k-stage cutting stock scenario.

Dealing with $k > 2$ stages requires the model to consider that a shelf can be opened by one or more items. This premise makes it necessary to generate a special type of items called dummy items. A dummy item can ether have a positive height with zero width or a positive width with zero height. In addition, the height or width of a dummy item is a combination of heights or widths of actual items. In what follows, some notation is introduced for the proposed model.

Sets and indexes

$l \in \{1,\ldots,p\}$: set of bins;

$i \in \{1,\ldots,n\}$: set of items;

$j \in \{n+1,\ldots,m\}$: set of dummy items;

$s \in \{1,\ldots,k\}$: set of stages;

$O(k) = \{3,5,\ldots,k\}$: set of odd stages when k is odd;

$O(k) = \{3,5,\ldots,k-1\}$: set of odd stages when k is even;

$E(k) = \{4,6,\ldots,k\}$: set of even stages when k is even;

$E(k) = \{4,6,\ldots,k-1\}$: set of even stages when k is odd.

Parameters

H_l: height of bin l;

W_l: width of bin l;

c_l: cost of bin l;

h_i: height of item type i;

w_i: width of item type i;

d_s: setup cost related to stage s;

k: maximum number of stages.

β_i: indexes of items sorted in terms of height such that $h_{\beta_1} \geqslant h_{\beta_2} \geqslant \ldots \geqslant h_{\beta_m}$;

γ_i: indexes of items sorted in terms of width such that $w_{\gamma_1} \geqslant w_{\gamma_2} \geqslant \ldots \geqslant w_{\gamma_m}$;

$$\alpha_i : \begin{cases} 1, & i = 1 \\ i, & h_{\beta_i} \neq h_{\beta_{i-1}} \forall i = 2,\ldots,m \\ \alpha_{i-1}, & h_{\beta_i} = h_{\beta_{i-1}} \forall i = 2,\ldots,m \end{cases}$$

$$\lambda_i : \begin{cases} 1, & i = 1 \\ i, & w_{\gamma_i} \neq w_{\gamma_{i-1}} \forall i = 2,\ldots,m \\ \lambda_{i-1}, & w_{\gamma_i} = w_{\gamma_{i-1}} \forall i = 2,\ldots,m \end{cases}$$

$$\theta_i : \begin{cases} m, & i = m \\ i, & h_{\beta_i} \neq h_{\beta_{i+1}} \forall i = m-1,\ldots,1 \\ \theta_{i+1}, & h_{\beta_i} = h_{\beta_{i+1}} \forall i = m-1,\ldots,1 \end{cases}$$

$$\delta_i : \begin{cases} m, & i = m \\ i, & w_{\gamma_i} \neq w_{\gamma_{i+1}} \forall i = m-1, \dots, 1 \\ \delta_{i+1}, & w_{\gamma_i} = w_{\gamma_{i+1}} \forall i = m-1, \dots, 1 \end{cases}$$

Decision variables

q_{jl}: 1, if item j opens a first stage shelf in bin l; and 0, otherwise;

$x^{(s)}_{ijl}$: 1, if item i opens a stage s shelf on stage $s-1$ shelf opened by item j on bin l; 0, otherwise;

u_l: 1, if bin l is used; 0, otherwise.

Objective function

$$\text{Minimize} \sum_{l=1}^{p} c_l u_l + \sum_{s=2}^{k} \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{l=1}^{p} d_s x^{(s)}_{ijl} \tag{3.1}$$

Constraints

$$\sum_{j=1}^{m} h_j q_{jl} \leqslant H_l u_l \quad \forall l \in \{1, \dots, p\} \tag{3.2}$$

$$\sum_{i=\alpha_j}^{m} w_{\beta_i} x^{(2)}_{\beta_i \beta_j l} \leqslant (W_l - w_{\beta_j}) q_{\beta_j l} \quad \forall j \in \{1, \dots, m\}, l \in \{1, \dots, p\} \tag{3.3}$$

$$\sum_{i=\lambda_j}^{m} h_{\gamma_i} x^{(s)}_{\gamma_i \gamma_j l} \leqslant \sum_{v=1}^{\theta_j} (h_{\beta_v} - h_{\gamma_j}) x^{(s-1)}_{\gamma_j \beta_v l} \quad \forall j \in \{1, \dots, m\}, l \in \{1, \dots, p\}, s \in O(k) \tag{3.4}$$

$$\sum_{i=\alpha_j}^{m} w_i x^{(s)}_{\beta_i \beta_j l} \leqslant \sum_{v=1}^{\delta_j} (w_{\gamma_v} - w_{\beta_j}) x^{(s-1)}_{\beta_j \gamma_v l} \quad \forall j \in \{1, \dots, m\}, l \in \{1, \dots, p\}, s \in E(k) \tag{3.5}$$

$$\sum_{l=1}^{p} q_{\beta_i l} + \sum_{s \in \{2\} \cup E(k)} \sum_{j=1}^{\theta_i} \sum_{l=1}^{p} x^{(s)}_{\beta_i \beta_j l} + \sum_{s \in O(k)} \sum_{j=1}^{\delta_v} \sum_{l=1}^{p} x^{(s)}_{\gamma_v \gamma_j l} = 1 \quad \forall i, v \in \{1, \dots, m \mid \beta_i = \gamma_v \leq n\} \tag{3.6}$$

$$\sum_{l=1}^{p} q_{\beta_i l} + \sum_{j=1}^{\theta_i} x^{(se)}_{\beta_i \beta_j l} + \sum_{j=1}^{\delta_v} x^{(so)}_{\gamma_v \gamma_j l} \leq 1 \quad \forall l \in \{1, \dots, p\}, se \in \{2\} \cup E(k), so \in O(k), \; i, v \in \{1, \dots, m \mid \beta_i = \gamma_v \geq n+1\} \tag{3.7}$$

$$q_{jl} \in \{0,1\} \quad \forall j \in \{1,\ldots,m\}, l \in \{1,\ldots,p\} \tag{3.8}$$

$$x_{ijl}^{(s)} \in \{0,1\} \quad \forall i,j \in \{1,\ldots,m\}, l \in \{1,\ldots,p\}, s \in \{2,\ldots,k\} \tag{3.9}$$

$$u_l \in \{0,1\} \quad \forall l \in \{1,\ldots,p\} \tag{3.10}$$

The objective function (3.1) minimizes the cost which is composed of bin usage cost and setup cost. Constraint set (3.2) ensures that, for each bin used, the sum of heights of the items that open a shelf on the first stage does not exceed the height of the associated bin. Constraint set (3.3) ensures that the sum of the widths of the items that open a second stage shelf on the first stage shelf opened by item j does not exceeds the width of the bin on which the shelves are been opened minus the width of the item j. Constraint set (3.4) ensures that, for each stage $s \in O(k)$, the sum of the heights of the items that open a stage s shelf at the stage $s-1$ shelf opened by item j does not exceed the height of the stage $s-2$ shelf on which item j is opened minus the height of the item j. Constraint set (3.5) ensures that, for each stage $s \in E(k)$, the sum of the widths of the items that open a stage s shelf at the stage $s-1$ shelf opened by item j does not exceed the width of the stage $s-2$ shelf on which item j is opened minus the width of the item j. Constraint set (3.6) ensures that the demand of each item is met. Constraint set 3.7 ensures that a dummy item can be used at most one time for each bin l and stage s. Constraint sets (3.8), (3.9) and (3.10) define the domain of the decision variables. The model has, $O(m^2pk)$ variables and $O(mpk)$ constraints.

The above-described model limits the number of cut stages to the parameter k, which means that the higher the value of k, the lower the value of the objective function. To guarantee the maximum possible number of cut stages, one can set $k = m$. The result is a model with $O(m^3p)$ variables and $O(m^2p)$ constraints.

Proposition 3.1 *If Z_k and Z_g are the global optimal solution values for the setup model in which k and g are the maximum number of stages allowed such that $k < g$, then $Z_k \geqslant Z_g$.*

Proof 3.1 *If Φ_k and Φ_g are the viable solution spaces for the setup model with k and g maximum number of stages such that $k < g$, it is known that $\Phi_k \subset \Phi_g$, hence $Z_k \geqslant Z_g$.*

Additionally, if all possible dummy items are considered, the total number of dummy items $(m-n)$ will be equal to $\sum_{i=2}^{n} C_i^n$ which is $O(2^n)$, resulting in a pseudo-polynomial model with $O(2^{3n}p)$ variables and $O(2^{2n}p)$ constraints.

Enumerating all possible dummy items can be prohibitive even for small-scale instances, therefore, we will consider a subset of dummy items generated as described in Algorithm 4. The enumeration considers all combinations with the elimination of repeated combinations. It is important to highlight that this dummy item subset does not guarantee optimality to the problem, i.e., the quality of the results depends on the method used to generate the dummy items.

Algorithm 4: Dummy items enumeration procedure

Data: H : array[p], W : array[p], h : array[n], w : array[n]

Result: m : integer, h : array[m], w : array[m]

```
maxH = max(H_l ∀ l ∈ {1,...,p});
maxW = max(W_l ∀ l ∈ {1,...,p});
m = n;
dh = [null], dw = [null];
for i = 1 to n − 1 do
    for j = i to n do
        x ← h_i + h_j, y ← w_i + w_j;
        if x ⩽ maxH and x ∉ dh then
            dh+ = x, m+ = 1;
        end
        if y ⩽ maxW and y ∉ dw then
            dw+ = y, m+ = 1;
        end
    end
end
stop = False;
N_1 = size(dh);
while not stop do
    for i = 1 to n do
        for j = 1 to N_1 do
            x = h_i + dh_j;
            if x ⩽ maxH and x ∉ dh then
                dh+ = x, m+ = 1;
            end
        end
    end
    if size(dh) > N_1 then
        N_1 = size(dh)
    else
        stop = True
    end
end
stop = False;
N_2 = size(dw);
while not stop do
    for i = 1 to n do
        for j = 1 to N_2 do
            y = w_i + dw_j;
            if y ⩽ maxW and y ∉ dw then
                dw+ = y, m+ = 1;
            end
        end
    end
    if size(dw) > N_2 then
        N_2 = size(dw)
    else
        stop = True
    end
end
h+ = dh;
w+ = [0] × N_1;
w+ = dw;
h+ = [0] × N_2;
```

In Algorithm 4, lines (1-2) set the maximum height and width of the bins, respectively. Line (4) defines the set of heights and widths of the dummy items as null. Lines (5-15) increment the set of heights and widths of the dummy items by combining the set of heights and widths of the actual items two-by-two, respectively. Lines (16-32) increment the set of heights of the dummy items combining the heights from the same set with the set of heights of the actual items until the set of heights of the dummy items is complete as stated in lines (27-31). Lines (33-49) increment the set of widths associated with the dummy items the same way as defined in lines (16-32) for the heights. Lines (50-53) update the set of heights and widths of the items with the set of heights and widths of the dummy items.

COMPUTATIONAL RESULTS

In this section we perform numerical experiments to analyze the setup model. Six special randomly generated small-scale instances were considered to asset the model (https://www.researchgate.net/publication/322702826_random_instances). It is important to highlight that medium to large-scale instances are impracticable to be solved by the model proposed and the instances provided by Andrade *et al.* (2016) did not allow more than two stages for most of the instances. The bins and items that compose each instance are described at Table 3, in which p is the number of bins and n is the number of items.

Table 3 – Instances

Instance	p	Objects (width x height)	n	Items (width x height)
1	2	15x10, 10x10	9	8x4, 3x7, 3x(8x2), 3x4, 2x(3x3), 2x1
2	3	40x40, 2x(40x70)	23	21x22, 31x13, 3x(9x35), 3x(9x24), 2x(30x7), 3x(11x13), 10x14, 3x(14x8), 3x(12x8), 3x(13x70)
3	4	70x70, 3x(100x100)	18	31x43, 30x41, 29x39, 28x38, 27x37, 2x(29x18), 2x(17x27), 15x24, 2x(16x25, 16x25), 2x(23x14), 2x(21x12), 19x11, 32x20
4	3	100x110, 80x80, 70x60	11	50x60, 50x40, 2x(20x50), 10x50, 25x20, 25x30, 2x(25x25), 65x10, 35x10
5	5	2x(70x70), 80x80, 80x90	13	2x(30x20), 3x(20x30), 4x(20x15), 10x30, 77x20, 40x10
6	4	2x(20x20), 2x(14x14)	22	4x(6x6), 2x(8x8), 4x(4x6), 6x(4x4), 6x(4x14)

For this numeric experiment we consider $c_l = H_l W_l (l = 1, \ldots, p)$ (i.e., cost is proportional to the bin area) and the setup cost is set as a proportion P of the smallest available bin area, i.e., $d_s = (s-2) \cdot P \cdot min\{H_l W_l \; \forall \; l = 1, \ldots, p\} (s = 2, \ldots, k)$. Additionally, a low and high setup cost configuration are taken into consideration, where $P = 5\%$ is the cost proportion for the low setup cost configuration and $P = 20\%$ is the cost proportion for the high setup cost configuration. To evaluate the impact of the maximum number of stages allowed (k) in the solution time, we consider $k = \{3, 4, 5\}$, resulting in six different configurations to be tested.

The models were implemented on python version 3.5 with the use of Pyomo mathematical modeling language (http://www.pyomo.org/) combined with IBM ILOG CPLEX 12.6.0 solver. The experiments were conducted on a machine with Intel® Core ™ i5-5200U CPU 2.20GHz x 4 processor, 4GB of RAM, running Ubuntu 16.04 LTS.

Table 4 present the number of dummy items generated by Algorithm 4 for each instance evaluated.

Table 4 – Dummy items

instance	number of items	number of dummy items	total number of items
1	9	19	28
2	23	76	99
3	18	145	163
4	11	33	44
5	13	19	32
6	22	14	36

Results are presented in Tables 5, 6 and 7 for $k = 3, 4$ and 5, respectively. In Tables 5 to 7, the first column indicates the instance which is being evaluated followed by the number of dummy items generated. The third column indicate the maximum number of stages used in the solution with low setup configuration. The fourth, fifth and seventh columns present the setup cost, bin cost and total cost (objective) with low setup configuration. Columns eight to twelve show the same information presented in columns three to seven but with respect to the hight setup cost configuration.

As expected, the use of higher number stages is inversely proportional to setup cost proportion. For $k = 3$ and a low setup cost configuration, only the first instance used the maximum number of stages allowed in contrast to a high setup cost configuration which had zero occurrence of three stages. For $k = 4$ and a low setup cost configuration, two instances used four stages and one used three stages. The results with a high setup cost configuration have one occurrence of four or three stages. Finally, for $k = 5$, the low setup configuration resulted in

16.6% of the solutions using five stages, 33.3% using four stages and 16.6% using three stages.

Furthermore, we evaluate the impact on the computational time when setup cost increases. On average, the computational time with low setup configuration took approximately 15 times more computational effort when compared to the high setup configuration results.

Table 5 – Computational results ($k = 3$)

inst.	low setup cost					high setup cost				
	max stages used	setup cost	bin cost	objective	solution time	max stages used	setup cost	bin cost	objective	solution time
1	3	30.0	150.0	180.0	0.19	2	0.0	250.0	250.0	0.17
2	2	0.0	5600.0	5600.0	2.86	2	0.0	5600.0	5600.0	1.44
3	2	0.0	14900.0	14900.0	7.08	2	0.0	14900.0	14900.0	5.27
4	2	0.0	15200.0	15200.0	0.26	2	0.0	15200.0	15200.0	0.32
5	2	0.0	11300.0	11300.0	1.06	2	0.0	11300.0	11300.0	0.48
6	2	0.0	996.0	996.0	6.10	2	0.0	996.0	996.0	3.65

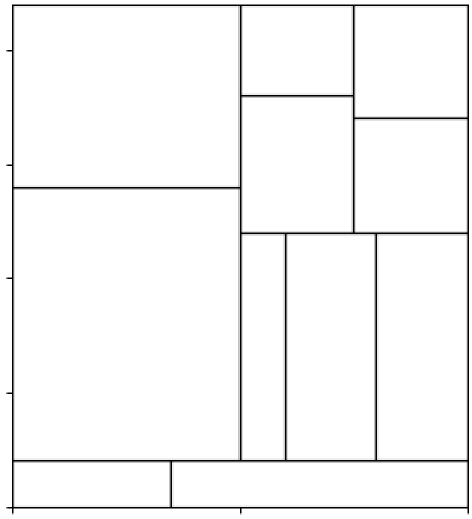
Table 6 – Computational results ($k = 4$)

inst.	low setup cost					high setup cost				
	max stages used	setup cost	bin cost	objective	solution time	max stages used	setup cost	bin cost	objective	solution time
1	3	30.0	150.0	180.0	0.48	2	0.0	250.0	250.0	0.33
2	2	0.0	5600.0	5600.0	6.98	2	0.0	5600.0	5600.0	3.03
3	2	0.0	14900.0	14900.0	20.99	2	0.0	14900.0	14900.0	14.80
4	2	0.0	15200.0	15200.0	3.74	2	0.0	15200.0	15200.0	1.14
5	4	2695.0	7200.0	9895.0	34.81	2	0.0	11300.0	11300.0	3.06
6	4	156.8	800.0	956.8	437.82	2	0.0	996.0	996.0	22.63

Table 7 – Computational results ($k = 5$)

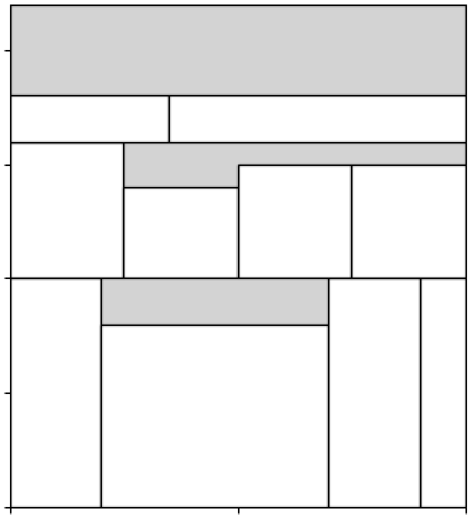
inst.	low setup cost					high setup cost				
	max stages used	setup cost	bin cost	objective	solution time	max stages used	setup cost	bin cost	objective	solution time
1	3	30.0	150.0	180.0	0.65	2	0.0	250.0	250.0	0.44
2	2	0.0	5600.0	5600.0	22.13	2	0.0	5600.0	5600.0	5.07
3	2	0.0	14900.0	14900.0	78.61	2	0.0	14900.0	14900.0	20.14
4	5	3570.0	11000.0	14570.0	8.20	2	0.0	15200.0	15200.0	2.27
5	4	2695.0	7200.0	9895.0	47.66	2	0.0	11300.0	11300.0	2.89
6	4	156.8	800.0	956.8	1457.54	2	0.0	996.0	996.0	23.35

Figure 11 – Solution for instance 4 with $k = 5$ and low setup configuration

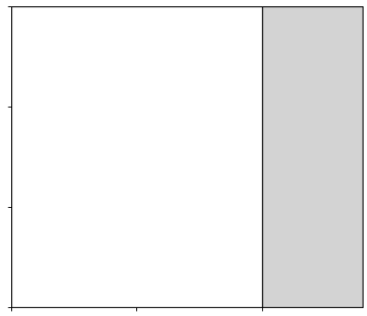


(a) Bin 1: 100 x 110

Figure 12 – Solution for instance 4 with $k = 5$ and high setup configuration



(a) Bin 1: 100 x 110



(b) Bin 2: 70 x 60

Figures 11 and 12 illustrate the solution for instance 4 with maximum number of stages $k = 5$ and low and high setup configuration, respectively. Due to the higher setup cost, the second case used a maximum of two stages of cut, resulting in the need for an extra bin.

CONCLUSIONS

In this research, a new variant of the two-dimensional guillotine cutting stock problem was presented. The variant dealt with the *k*-stage case in which setup cost related to the stages is considered. To the best of our knowledge, this research is the first to deal with the new variant. The problem emerged from the granite industry during technical visits to a local manufacturer. A pseudo-polynomial mathematical formulation with $O(2^{3n}p)$ variables and $O(2^{2n}p)$ constraints was presented and numerical experiments were conducted with small-scale randomly generated instances using the MIP solver IBM ILOG CPLEX. The model was able to solve with optimality all instances tested in acceptable computational time when a limited number of dummy items was generated. However, to guarantee the global optimal solution, one must enumerate all possible dummy items, which is impracticable even for small scale instances. Moreover, the model can be used to find good initial solutions for heuristic approaches by adding dummy items according to the quality desired for the solution. This concept, to the best of our knowledge, have not being explored by the revised literature.

Future work should concentrate on a column generation procedure for the dummy items since the approach would eliminate the need for their full enumeration. Additionally, our approach incorporates the setup cost as a penalty in the objective function. However, the problem in study has the property of being multi-objective; hence, this property must be taken into consideration in future works.

REFERENCES

ALVAREZ-VALDES, R.; PARAJON, A.; TAMARIT, J. M. A computational study of lp-based heuristic algorithms for two-dimensional guillotine cutting stock problems. **OR Spectrum**, Springer, v. 24, n. 2, p. 179–192, 2002.

ALVAREZ-VALDÉS, R.; PARREÑO, F.; TAMARIT, J. M. A branch-and-cut algorithm for the pallet loading problem. **Computers & Operations Research**, Elsevier, v. 32, n. 11, p. 3007–3029, 2005.

ANDRADE, R.; BIRGIN, E.; MORABITO, R. Two-stage two-dimensional guillotine cutting stock problems with usable leftover. **International Transactions in Operational Research**, Wiley Online Library, v. 23, n. 1-2, p. 121–145, 2016.

BELOV, G.; SCHEITHAUER, G. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. **European journal of operational research**, Elsevier, v. 171, n. 1, p. 85–106, 2006.

BIRGIN, E. G.; LOBATO, R. D. Orthogonal packing of identical rectangles within isotropic convex regions. **Computers & Industrial Engineering**, Elsevier, v. 59, n. 4, p. 595–602, 2010.

BIRGIN, E. G.; LOBATO, R. D.; MORABITO, R. An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. **Journal of the Operational Research Society**, Springer, v. 61, n. 2, p. 306–320, 2010.

BIRGIN, E. G.; MORABITO, R.; NISHIHARA, F. H. A note on an l-approach for solving the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, Springer, v. 56, n. 12, p. 1448–1451, 2005.

BORTFELDT, A.; JUNGMANN, S. A tree search algorithm for solving the multi-dimensional strip packing problem with guillotine cutting constraint. **Annals of Operations Research**, Springer, v. 196, n. 1, p. 53–71, 2012.

CHRISTOFIDES, N.; HADJICONSTANTINOU, E. An exact algorithm for orthogonal 2-d cutting problems using guillotine cuts. **European Journal of Operational Research**, Elsevier, v. 83, n. 1, p. 21–38, 1995.

CINTRA, G.; MIYAZAWA, F. K.; WAKABAYASHI, Y.; XAVIER, E. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. **European Journal of Operational Research**, Elsevier, v. 191, n. 1, p. 61–85, 2008.

CUI, Y.; CUI, Y.-P.; YANG, L. Heuristic for the two-dimensional arbitrary stock-size cutting stock problem. **Computers & Industrial Engineering**, Elsevier, v. 78, p. 195–204, 2014.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, Elsevier, v. 44, n. 2, p. 145–159, 1990.

FURINI, F.; MALAGUTI, E. Models for the two-dimensional two-stage cutting stock problem with multiple stock size. **Computers & Operations Research**, Elsevier, v. 40, n. 8, p. 1953–1962, 2013.

FURINI, F.; MALAGUTI, E.; THOMOPULOS, D. Modeling two-dimensional guillotine cutting problems via integer programming. **INFORMS Journal on Computing**, INFORMS, v. 28, n. 4, p. 736–751, 2016.

GHANDFOROUSH, P.; DANIELS, J. J. A heuristic algorithm for the guillotine constrained cutting stock problem. **ORSA journal on Computing**, INFORMS, v. 4, n. 3, p. 351–356, 1992.

GILMORE, P.; GOMORY, R. E. Multistage cutting stock problems of two and more dimensions. **Operations research**, INFORMS, v. 13, n. 1, p. 94–120, 1965.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. **Operations research**, INFORMS, v. 9, n. 6, p. 849–859, 1961.

HIFI, M. Exact algorithms for the guillotine strip cutting/packing problem. **Computers & Operations Research**, Elsevier, v. 25, n. 11, p. 925–940, 1998.

HIFI, M.; NEGRE, S.; OUAFI, R.; SAADI, T. A parallel algorithm for constrained two-staged two-dimensional cutting problems. **Computers & Industrial Engineering**, Elsevier, v. 62, n. 1, p. 177–189, 2012.

LINS, L.; LINS, S.; MORABITO, R. An l-approach for packing (l, w)-rectangles into rectangular and l-shaped pieces. **Journal of the Operational Research Society**, Springer, v. 54, n. 7, p. 777–789, 2003.

LODI, A.; MONACI, M.; PIETROBUONI, E. Partial enumeration algorithms for two-dimensional bin packing problem with guillotine constraints. **Discrete Applied Mathematics**, Elsevier, 2015.

MACLEOD, B.; MOLL, R.; GIRKAR, M.; HANIFI, N. An algorithm for the 2d guillotine cutting stock problem. **European Journal of Operational Research**, Elsevier, v. 68, n. 3, p. 400–412, 1993.

MORABITO, R.; FARAGO, R. A tight lagrangean relaxation bound for the manufacturer's pallet loading problem. **Stud. Inform. Univ.**, v. 2, n. 1, p. 57–76, 2002.

MORABITO, R.; MORALES, S. A simple and effective recursive procedure for the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, Palgrave Macmillan, v. 49, n. 8, p. 819–828, 1998.

MORABITO, R.; PUREZA, V. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. **Annals of Operations Research**, Springer, v. 179, n. 1, p. 297–315, 2010.

NTENE, N.; VUUREN, J. H. van. A survey and comparison of guillotine heuristics for the 2d oriented offline strip packing problem. **Discrete Optimization**, Elsevier, v. 6, n. 2, p. 174–188, 2009.

PUREZA, V.; MORABITO, R. Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. **Computers & Operations Research**, Elsevier, v. 33, n. 3, p. 804–819, 2006.

RAM, B. The pallet loading problem: A survey. **International Journal of Production Economics**, Elsevier, v. 28, n. 2, p. 217–225, 1992.

RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. **Computers & Operations Research**, Elsevier, v. 34, n. 9, p. 2695–2708, 2007.

SILVA, E.; ALVELOS, F.; CARVALHO, J. V. de. An integer programming model for two-and three-stage two-dimensional cutting stock problems. **European Journal of Operational Research**, Elsevier, v. 205, n. 3, p. 699–708, 2010.

SONG, X.; CHU, C.; NIE, Y. A heuristic dynamic programming algorithm for 2d unconstrained guillotine cutting. **WSEAS Transactions on Mathematics**, v. 3, p. 230–238, 2004.

STEUDEL, H. J. Generating pallet loading patterns: a special case of the two-dimensional cutting stock problem. **Management Science**, INFORMS, v. 25, n. 10, p. 997–1004, 1979.

SULIMAN, S. A sequential heuristic procedure for the two-dimensional cutting-stock problem. **International Journal of Production Economics**, Elsevier, v. 99, n. 1, p. 177–185, 2006.

VANDERBECK, F. A nested decomposition approach to a three-stage, two-dimensional cutting-stock problem. **Management Science**, INFORMS, v. 47, n. 6, p. 864–879, 2001.

WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European journal of operational research**, Elsevier, v. 183, n. 3, p. 1109–1130, 2007.

WEI, L.; TIAN, T.; ZHU, W.; LIM, A. A block-based layer building approach for the 2d guillotine strip packing problem. **European Journal of Operational Research**, Elsevier, v. 239, n. 1, p. 58–69, 2014.

YOUNG-GUN, G.; KANG, M.-K. A fast algorithm for two-dimensional pallet loading problems of large size. **European Journal of Operational Research**, Elsevier, v. 134, n. 1, p. 193–202, 2001.

4 CONCLUSIONS AND FUTURE WORKS

In this work, we explored two new variants of the two-dimensional guillotine multiple stock size cutting stock problem. The variants were identified during technical visits to a local granite beneficiary. Firstly, we dealt with the two-stage two-dimensional guillotine cutting stock problem in which the decision is to choose the size of the identical items. Two mathematical models and two solution procedures were presented to solve the non-rotational and rotational cases. The models and procedures were limited to the case in which the set of possible item sizes is discrete (i.e. there is a limited number of item types). Numerical experiments were conducted on 40 randomly generated instances. The computational results showed the efficiency of the approach especially for the case without orthogonal rotation of items, since the instances tested were easily solved to optimality. Finally, we dealt with the k-stage two-dimensional guillotine cutting stock problem in which setup cost associated with stages is considered. An $O(2^n p)$ mathematical formulation was presented, in which n is the number of items and p is the number of bins, as an extension of the bin packing model M_1 of Andrade *et al.* (2016). The model can be configured to find good viable solutions through the generation of dummy items. Numerical experiments were conducted on 6 arbitrary small-scale randomly generated instances. All instances tested were solved to optimality with acceptable computational time for a limited viable space; however, by taking into consideration the high order of complexity, the approach would be impracticable for larger instances.

To the best of our knowledge, there are no other studies in the available literature dealing with the above mentioned problems and, due to the limitations highlighted, future studies on the current topic are required. On the first variant, the case in which the height and width of the identical items are a continuous variable must be investigated, since the discrete premise limits the solution space. On the second variant, more computationally efficient approaches must be explored. Furthermore, the second variant has a multi-objective characteristic that has not been taken into consideration by this research. Hence, this approach must be investigated in future works.

REFERENCES

ALVAREZ-VALDES, R.; PARAJON, A.; TAMARIT, J. M. A computational study of lp-based heuristic algorithms for two-dimensional guillotine cutting stock problems. **OR Spectrum**, Springer, v. 24, n. 2, p. 179–192, 2002.

ALVAREZ-VALDÉS, R.; PARREÑO, F.; TAMARIT, J. M. A branch-and-cut algorithm for the pallet loading problem. **Computers & Operations Research**, Elsevier, v. 32, n. 11, p. 3007–3029, 2005.

ANDRADE, R.; BIRGIN, E.; MORABITO, R. Two-stage two-dimensional guillotine cutting stock problems with usable leftover. **International Transactions in Operational Research**, Wiley Online Library, v. 23, n. 1-2, p. 121–145, 2016.

BELOV, G.; SCHEITHAUER, G. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. **European journal of operational research**, Elsevier, v. 171, n. 1, p. 85–106, 2006.

BIRGIN, E. G.; LOBATO, R. D. Orthogonal packing of identical rectangles within isotropic convex regions. **Computers & Industrial Engineering**, Elsevier, v. 59, n. 4, p. 595–602, 2010.

BIRGIN, E. G.; LOBATO, R. D.; MORABITO, R. An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. **Journal of the Operational Research Society**, Springer, v. 61, n. 2, p. 306–320, 2010.

BIRGIN, E. G.; MORABITO, R.; NISHIHARA, F. H. A note on an l-approach for solving the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, Springer, v. 56, n. 12, p. 1448–1451, 2005.

BORTFELDT, A.; JUNGMANN, S. A tree search algorithm for solving the multi-dimensional strip packing problem with guillotine cutting constraint. **Annals of Operations Research**, Springer, v. 196, n. 1, p. 53–71, 2012.

CHRISTOFIDES, N.; HADJICONSTANTINOU, E. An exact algorithm for orthogonal 2-d cutting problems using guillotine cuts. **European Journal of Operational Research**, Elsevier, v. 83, n. 1, p. 21–38, 1995.

CINTRA, G.; MIYAZAWA, F. K.; WAKABAYASHI, Y.; XAVIER, E. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. **European Journal of Operational Research**, Elsevier, v. 191, n. 1, p. 61–85, 2008.

CUI, Y.; CUI, Y.-P.; YANG, L. Heuristic for the two-dimensional arbitrary stock-size cutting stock problem. **Computers & Industrial Engineering**, Elsevier, v. 78, p. 195–204, 2014.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, Elsevier, v. 44, n. 2, p. 145–159, 1990.

FURINI, F.; MALAGUTI, E. Models for the two-dimensional two-stage cutting stock problem with multiple stock size. **Computers & Operations Research**, Elsevier, v. 40, n. 8, p. 1953–1962, 2013.

FURINI, F.; MALAGUTI, E.; THOMOPULOS, D. Modeling two-dimensional guillotine cutting problems via integer programming. **INFORMS Journal on Computing**, INFORMS, v. 28, n. 4, p. 736–751, 2016.

GHANDFOROUSH, P.; DANIELS, J. J. A heuristic algorithm for the guillotine constrained cutting stock problem. **ORSA journal on Computing**, INFORMS, v. 4, n. 3, p. 351–356, 1992.

GILMORE, P.; GOMORY, R. E. Multistage cutting stock problems of two and more dimensions. **Operations research**, INFORMS, v. 13, n. 1, p. 94–120, 1965.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. **Operations research**, INFORMS, v. 9, n. 6, p. 849–859, 1961.

HIFI, M. Exact algorithms for the guillotine strip cutting/packing problem. **Computers & Operations Research**, Elsevier, v. 25, n. 11, p. 925–940, 1998.

HIFI, M.; NEGRE, S.; OUAFI, R.; SAADI, T. A parallel algorithm for constrained two-staged two-dimensional cutting problems. **Computers & Industrial Engineering**, Elsevier, v. 62, n. 1, p. 177–189, 2012.

LINS, L.; LINS, S.; MORABITO, R. An l-approach for packing (l, w)-rectangles into rectangular and l-shaped pieces. **Journal of the Operational Research Society**, Springer, v. 54, n. 7, p. 777–789, 2003.

LODI, A.; MONACI, M.; PIETROBUONI, E. Partial enumeration algorithms for two-dimensional bin packing problem with guillotine constraints. **Discrete Applied Mathematics**, Elsevier, 2015.

MACLEOD, B.; MOLL, R.; GIRKAR, M.; HANIFI, N. An algorithm for the 2d guillotine cutting stock problem. **European Journal of Operational Research**, Elsevier, v. 68, n. 3, p. 400–412, 1993.

MORABITO, R.; FARAGO, R. A tight lagrangean relaxation bound for the manufacturer's pallet loading problem. **Stud. Inform. Univ.**, v. 2, n. 1, p. 57–76, 2002.

MORABITO, R.; MORALES, S. A simple and effective recursive procedure for the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, Palgrave Macmillan, v. 49, n. 8, p. 819–828, 1998.

MORABITO, R.; PUREZA, V. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. **Annals of Operations Research**, Springer, v. 179, n. 1, p. 297–315, 2010.

NTENE, N.; VUUREN, J. H. van. A survey and comparison of guillotine heuristics for the 2d oriented offline strip packing problem. **Discrete Optimization**, Elsevier, v. 6, n. 2, p. 174–188, 2009.

PUREZA, V.; MORABITO, R. Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. **Computers & Operations Research**, Elsevier, v. 33, n. 3, p. 804–819, 2006.

RAM, B. The pallet loading problem: A survey. **International Journal of Production Economics**, Elsevier, v. 28, n. 2, p. 217–225, 1992.

RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. **Computers & Operations Research**, Elsevier, v. 34, n. 9, p. 2695–2708, 2007.

SILVA, E.; ALVELOS, F.; CARVALHO, J. V. de. An integer programming model for two-and three-stage two-dimensional cutting stock problems. **European Journal of Operational Research**, Elsevier, v. 205, n. 3, p. 699–708, 2010.

SONG, X.; CHU, C.; NIE, Y. A heuristic dynamic programming algorithm for 2d unconstrained guillotine cutting. **WSEAS Transactions on Mathematics**, v. 3, p. 230–238, 2004.

STEUDEL, H. J. Generating pallet loading patterns: a special case of the two-dimensional cutting stock problem. **Management Science**, INFORMS, v. 25, n. 10, p. 997–1004, 1979.

SULIMAN, S. A sequential heuristic procedure for the two-dimensional cutting-stock problem. **International Journal of Production Economics**, Elsevier, v. 99, n. 1, p. 177–185, 2006.

VANDERBECK, F. A nested decomposition approach to a three-stage, two-dimensional cutting-stock problem. **Management Science**, INFORMS, v. 47, n. 6, p. 864–879, 2001.

WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European journal of operational research**, Elsevier, v. 183, n. 3, p. 1109–1130, 2007.

WEI, L.; TIAN, T.; ZHU, W.; LIM, A. A block-based layer building approach for the 2d guillotine strip packing problem. **European Journal of Operational Research**, Elsevier, v. 239, n. 1, p. 58–69, 2014.

YOUNG-GUN, G.; KANG, M.-K. A fast algorithm for two-dimensional pallet loading problems of large size. **European Journal of Operational Research**, Elsevier, v. 134, n. 1, p. 193–202, 2001.