Journal of Mathematical Modeling
Vol. 13, No. 2, 2025, pp. 415-428. Research Article

JMM

A polynomial mixed-integer linear programming model for two-dimensional guillotine strip packing problem

Ashkan Fakhri *

Department of Mathematics, University of Science and Technology of Mazandaran, P.O. Box 4851878195, Behshahr, Iran
Email(s): a.fakhri@mazust.ac.ir; fakhriashkan@gmail.com

Abstract. A two-dimensional strip packing problem is the process of packing a set of rectangular items of given dimensions into a strip of bounded width and infinite height so that the used height of the strip is minimized. In the case that only guillotine packing is permitted, the problem is called the guillotine strip packing problem (GSPP). Guillotine packing commonly arises in different industries such as glass, steel, paper and wood. Nevertheless, there is a lack of explicit mathematical models for GSPP that can globally solve the problem. In this paper, a new mixed-integer programming model inspired by a so-called sequence sub-tour elimination technique for the traveling salesman problem (TSP) is presented as a relaxation of (non-staged) GSPP with orthogonal rotation. The proposed model is able to find good solutions (good upper bounds) for the optimal objective value and more importantly, it is a polynomial model of order $O(n^2)$, i.e. the number of decision variables (and constraints, as well) is a polynomial of order $O(n^2)$ in the number of the rectangular items (n). Numerical results show that the solutions obtained from the proposed model are superior to several existing heuristic algorithms in the literature.

Keywords: Packing, cutting, guillotine cut, polynomial model, mixed-integer programming.
AMS Subject Classification 2010: 90C27, 90C11.

1 Introduction

Two-Dimensional Strip Packing Problem (2D-SPP) is the problem of determining the best way of packing specified smaller rectangular items into a single bin (strip) of bounded width and infinite height so that the used height of the strip is minimized. The problem is strongly NP-hard and it is well-known in production planning for a number of industries such as glass, steel, paper, newspaper printing, wood, transportation (to load containers), etc. [14]. In this paper, we assume that only orthogonal cuts are considered i.e. sides of each item are needed to be parallel to the sides of strip. While in the basic form of the problem, the items are assumed to have fixed orientation, orthogonal (90-degree) rotation of the items is

*Corresponding author
Received: 4 September 2024 / Revised: 29 November 2024 / Accepted: 11 December 2024
DOI: 10.22124/jmm.2024.26164.2320

allowed in our model. A guillotine (packing) pattern is a pattern that its items can be obtained through a series of guillotine cuts (also called edge-to-edge cuts) on the strip. This type of cuts is common in several industries such as the glass, wood and paper industries. Clearly, imposing guillotine constraints may lead to worse optimal patterns. If the total number of horizontal and vertical guillotine cuts is limited by a constant k, k-staged strip packing problem is obtained. Note that in this case, all parallel cuts which can be performed simultaneously are counted one time. The problem is called level packing when $k = 2$. In several papers different heuristics and approximation algorithms are developed for two-dimensional packing/cutting problems, including on-line algorithms [9], constructive heuristics [2, 5], set covering based heuristics [11, 25], local search [1, 33] and metaheuristics [15, 16, 28, 31] among others. Also, several relaxation techniques have been proposed to provide (lower/upper) bounds for these problems [3, 4]. For a relatively comprehensive survey on heuristics and relaxations, one can refer to [14] and the references therein. Heuristic approaches can obtain a feasible solution in a reasonable time, however there are few articles in the literature in which the achievement of an optimal solution is considered. Exact methods are, actually, Mixed-Integer Linear Programming (MILP) formulations that are solved using suitable MILP solvers. These formulations can be classified into three categories: polynomial, pseudo-polynomial and exponential models.

- Exponential models

Exponential models for two-dimensional packing/cutting problems can be divided into two types. Some of them relate variables to feasible patterns which leads to an exponential number of variables and they usually are implemented through a column generation framework. Some others are based on Benders' decomposition (leading an exponential number of constraints). To review this class of models, see [14] and the references therein.

- Pseudo-polynomial models

Pseudo-polynomial models relate variables to the positions into the bins where the items can be packed. Hence the number of decision variables and constraints depend on the discretization of the packing area. In such models, decision variables are usually related to the bottom-left corner of an item in the pattern. Several families of pseudo-polynomial models are proposed in the literature, including Arc-flow formulation [20, 26], One-cut models [12, 32] and Scheduling inspired models [6, 7] among others.

- Polynomial models

Polynomial models for packing problems are those that associate variables to the items. In such models, the number of decision variables (as well as the number of constraints) is a polynomial in the number of rectangular items. The authors in [8] considered the possible positions of each pair of items relative to each other and presented a polynomial mixed-integer model for the two-dimensional packing problem. A slight modification of the polynomial models of Chen et al. was presented in [17]. These polynomial models consider, in fact, the assortment problem (i.e., the case in which the number of bins is limited to one). Huang et al. [13] extended the mentioned polynomial models to a general 2D-CSP . None of the mentioned polynomial models is restricted to guillotine cuts. Guillotine constraints were proposed for level packing problems [18] and 2-staged knapsack problems [19]. These models were extended to the 3-staged case of the 2D-BPP [29] and to the 2-staged 2D-CSP with variable-sized bins [10]. Necessary and sufficient conditions for characterizing guillotine patterns were proposed in [23] and based on this characterization, the guillotine constraints were formulated as linear inequalities. However, the model of [23] is of order $O(n^4)$ where n is the number of the rectangular items. Having too many binary variables and constraints, this model is impractical and only has been tested on very small-sized instances ($n = 5$). In [22] two-dimensional guillotine placement problem was addressed and pseudo-polynomial and compact integer non-linear formulations were proposed for the problem and then equivalent MILP formulations were presented. Very recently, [30] proposed a compact mixed-integer formulation (which is a general heuristic model) for 2D guillotine cutting problems. Computational experiments were conducted comparing the proposed model with the 2D guillotine single knapsack problem and the 2D guillotine cutting stock and bin packing problems. As a limitation, the proposed model is not able to include some patterns. To overcome this weakness, an exact version of the model was proposed by the same authors, but the exact version (although general and theoretically interesting) is not computationally competitive.

In the present paper, we propose an $O(n^2)$ polynomial MILP relaxation of the GSPP. Although our model does not guarantee the achievement of optimal patterns, it can be applied in problems much larger in size compared to [23]. Indeed, numerical results show that the solutions obtained from our model are superior to several existing heuristic algorithms in the literature for not too large instances. Furthermore, our model has the advantage over previous k-staged mathematical models (such as [18, 19] and [29]) that it is not limited to 2-staged and 3-staged patterns. The rest of the paper is organized as follows: The proposed polynomial MILP relaxation of the GSPP is presented in Section 2. Section 3 reports the results of a numerical experiment and certifies the performance of the presented model. Finally, Section 4 concludes the paper and provides some directions to the future extensions.

2 The MILP relaxation

In this section, we aim to propose a mathematical model that is a relaxation of the GSPP. First we present a polynomial non-guillotine strip packing problem (NGSPP) and then we modify it to exclude non-guillotine patterns.

2.1 An MILP model for NGSPP

Assume that n items must be packed on a strip of width W and infinite height using a minimum length of the strip. To mathematically formulate the problem, we use the following parameters and decision variables, as they were used in [8] and [17]:

Parameters:

n: The number of given rectangular items to be packed.

(p_i, q_i): The width and length of a given rectangular item i for $i = 1, 2, \ldots, n$.

$\bar{x}$: The width of the strip.

$\bar{y}$: An upper bound of the required length of the strip. It can be simply chosen as $\bar{y} = \sum_{i=1}^{n} q_i$ if no better bound is known.

I: The set of indices of rectangular items ($I = \{1, 2, ..., n\}$).

Decision variables:

(x_i, y_i): The bottom-left coordinates of rectangular item i.

$(a_{ij}, b_{ij}, c_{ij}, d_{ij})$: A set of binary variables used to express the non-overlapping conditions for a pair of items i and j.

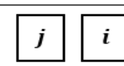
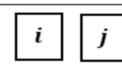
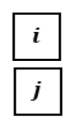
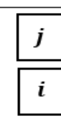
s_i: A binary variable which indicates orientation of item i. $s_i = 1$ if the side with size p_i of rectangular item i is parallel to the x-axis; otherwise, $s_i = 0$.

Y: The used length of the strip.

Using the above notation, the NGSPP can be expressed as:

$$
\begin{aligned}
& \textit{Minimize} \quad Y && && (1)\\
& s.t. && && \\
& x_j + p_j s_j + q_j(1-s_j) \le x_i + \bar{x}(1-a_{ij}) && i,j \in I,\ i<j, && (2)\\
& x_i + p_i s_i + q_i(1-s_i) \le x_j + \bar{x}(1-b_{ij}) && i,j \in I,\ i<j, && (3)\\
& y_j + q_j s_j + p_j(1-s_j) \le y_i + \bar{y}(1-c_{ij}) && i,j \in I,\ i<j, && (4)\\
(SPP) \quad & y_i + q_i s_i + p_i(1-s_i) \le y_j + \bar{y}(1-d_{ij}) && i,j \in I,\ i<j, && (5)\\
& a_{ij} + b_{ij} + c_{ij} + d_{ij} = 1 && i,j \in I,\ i<j, && (6)\\
& x_i + p_i s_i + q_i(1-s_i) \le \bar{x} && i \in I, && (7)\\
& y_i + q_i s_i + p_i(1-s_i) \le Y && i \in I, && (8)\\
& x_i \ge 0, y_i \ge 0, s_i \in \{0,1\} && i \in I, && (9)\\
& a_{ij}, b_{ij}, c_{ij}, d_{ij} \in \{0,1\} && i,j \in I,\ i<j. && (10)
\end{aligned}
$$

Remark 1. *Constraints (2)-(5) prevent overlapping of each pair of distinct items i and j. In fact, based on the values of a_{ij}, b_{ij}, c_{ij} and d_{ij}, one of the following cases can occur for each rectangular items i and j:*

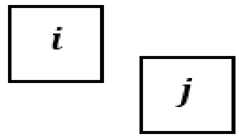
Case	a_{ij}	b_{ij}	c_{ij}	d_{ij}	Condition	Schematic Condition
1	1	0	0	0	i is located to the right of j	
2	0	1	0	0	i is located to the left of j	
3	0	0	1	0	i is located above j	
4	0	0	0	1	i is located to under j	

2.2 Sufficient conditions for excluding non-guillotine patterns

In order to derive sufficient conditions for guaranteeing the patterns to be guillotinable, we propose to use a sequence idea inspiring the idea of [24] for sub-tour elimination in the Traveling Salesman Problem (TSP).

Definition 1. *An edge e is called an obstacle to another edge e′ if the guillotine cut along e′ passes through e, and e is the closest edge to e′ (among the edges of the object to which it belongs) that has this property.*

For example, in the following figure, the right edge of i is an obstacle for the top edge of j (while the left edge of i is not an obstacle). Furthermore, the left edge of j is an obstacle for the bottom edge of i (while the right edge of j is not an obstacle).



Now, corresponding to each edge of a rectangular item $i \in I$, we consider a scalar label as follows:
z_{il}: the label related to the left edge of item i;
z_{ir}: the label related to the right edge of item i;
z_{ib}: the label related to the bottom edge of item i;
z_{it}: the label related to the top edge of item i.

Lemma 1. *Consider a feasible solution (x,s,a,b,c,d) of the problem SPP defined by (1)-(10). Then, the following two statements are equivalent:*
(s1) *There exists a set $z = \{z_{il}, z_{ir}, z_{it}, z_{ib} \ : \ i \in I\}$ of labels that satisfy the following conditions for each distinct items i and j in I:*

$$\textbf{(i)} \quad \textit{if} \quad x_j < x_i < x_j + p_j s_j + q_j(1-s_j) \quad \textit{then} \quad z_{il} > z_{jt} c_{(ij)} + z_{jb} d_{(ij)}, \tag{11}$$

$$\textbf{(ii)} \quad \textit{if} \quad x_j < x_i + p_i s_i + q_i(1-s_i) < x_j + p_j s_j + q_j(1-s_j) \quad \textit{then} \quad z_{ir} > z_{jt} c_{(ij)} + z_{jb} d_{(ij)}, \tag{12}$$

$$\textbf{(iii)} \quad \textit{if} \quad y_j < y_i < y_j + q_j s_j + p_j(1-s_j) \quad \textit{then} \quad z_{ib} > z_{jl} b_{(ij)} + z_{jr} a_{(ij)}, \tag{13}$$

$$\textbf{(iv)} \quad \textit{if} \quad y_j < y_i + q_i s_i + p_i(1-s_i) < y_j + q_j s_j + p_j(1-s_j) \quad \textit{then} \quad z_{it} > z_{jl} b_{(ij)} + z_{jr} a_{(ij)}, \tag{14}$$

where,

$$a_{(ij)} := \begin{cases} a_{ij} & i < j \\ b_{ji} & j < i \end{cases}, \tag{15}$$

$$b_{(ij)} := \begin{cases} b_{ij} & i < j \\ a_{ji} & j < i \end{cases}, \tag{16}$$

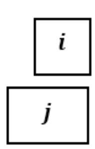
$$c_{(ij)} := \begin{cases} c_{ij} & i < j \\ d_{ji} & j < i \end{cases}, \tag{17}$$

$$d_{(ij)} := \begin{cases} d_{ij} & i < j \\ c_{ji} & j < i \end{cases}. \tag{18}$$

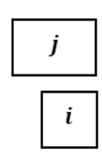
(s2) *For each arbitrary couple of edges e and e' in the pattern related to (x,s,a,b,c,d), if e is an obstacle to e' then e takes a smaller label than e'.*

Proof. Note that the if-clauses in the conditions (i)-(iv) cover all possible situations that an arbitrary edge e (belong to an arbitrary item j) is an obstacle to another edge e' (belong to an arbitrary item i). To prove that (s1) implies (s2), it is sufficient to show that the then-clauses guarantee that in all such situations, e takes a smaller label than e'. Condition (i) deals with the situations that either the top edge or the bottom edge of item j is an obstacle for the left edge of item i:

(i-a):



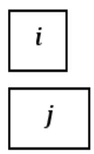
(i-b):



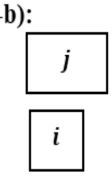
In the case (i-a), we have either $c_{(ij)} = c_{ij} = 1$ and $d_{(ij)} = d_{ij} = 0$ (if $i < j$) or $c_{(ij)} = d_{ji} = 1$ and $d_{(ij)} = c_{ji} = 0$ (if $i > j$); and hence, Condition (i) guarantees that the label of the left edge of the item i must be greater than the label of the top edge of the item j i.e. $z_{il} > z_{jt}$. Similarly, in the case (i-b), we have $c_{(ij)} = c_{ij} = 0$ and $d_{(ij)} = d_{ij} = 1$ (if $i < j$) or $c_{(ij)} = d_{ji} = 0$ and $d_{(ij)} = c_{ji} = 1$ (if $i > j$); and hence, Condition (i) guarantees that the label of the left edge of i must be greater than the label of the bottom edge of j i.e. $z_{il} > z_{jb}$. Similar discussions can be made for Conditions (ii)-(iv). Hence, (s1) implies (s2). The proof of the reverse direction is quite straightforward. □

Remark 2. *Condition (ii), presented in (12), deals with the situations that either the top edge or the bottom edge of item j is an obstacle to the right edge of item i:*

(ii-a):

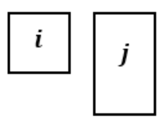


(ii-b):

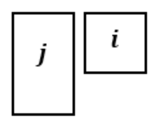


Condition (iii) deals with the situations that either the left edge or the right edge of item j is an obstacle to the bottom edge of item i:

(iii-a):

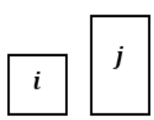


(iii-b):

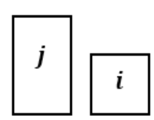


Finally, Condition (iv) deals with the situations that either the left edge or the right edge of item j is an obstacle to the top edge of item i:

(iv-a):



(iv-b)



Remark 3. *Statement (s2) states a sufficient condition for the label of e to be smaller than the label of e′. However, the reverse is not necessary i.e. if e takes a smaller label than e′, it is not necessary for e to be an obstacle to e′.*

Proposition 1. *Assume that there exists a set $z = \{z_{il}, z_{ir}, z_{it}, z_{ib} \ : \ i \in I\}$ of labels so that the conditions (i)-(iv) (defined in (11)-(14)) are satisfied for all $i, j \in I$ (and $i \neq j$). Then the pattern derived from the problem SPP defined by (1)-(10) is a guillotine pattern.*

Proof. By contradiction, assume that there exists a non-guillotine pattern with a label set z satisfying the conditions (i)-(iv). This pattern clearly includes a sub-pattern S on which no guillotine cut can be made. By sub-pattern we mean a subset of rectangular items retaining their location in the pattern. Consider an item i_1 in S having an edge e_1 with a label $z_{i_1\alpha_1}$ that must be cut (α_1 is a representative for one of the notations l, r, t or b). Since we assumed that no guillotine cut can be made on S, there is another item $i_2 \in S$ having an edge e_2 with a label $z_{i_2\alpha_2}$ which is an obstacle to e_1. If we continue this argument, we conclude that there is a sequence of edges e_1, e_2, e_3, e_4, that each one is an obstacle to the previous one. Clearly, there is some edge e_k in this sequence that is identical to a preceding edge $e_{k'}$ (with $k' < k$) because the number of items in S is finite. Since (according to Lemma 1) each edge in the sequence has a smaller label than the previous one, we have:$z_{i_{k'}\alpha_{k'}} > z_{i_{k'+1}\alpha_{k'+1}} > \cdots > z_{i_k\alpha_k} = z_{i_{k'}\alpha_{k'}}$ This contradiction approves that non-guillotine patterns cannot provide any label sets that satisfy conditions (i)-(iv). □

To illustrate more clearly how the conditions (i)-(iv) do not accept non-guillotine patterns, we present the following example.

Example 1. Consider the following non-guillotine pattern:

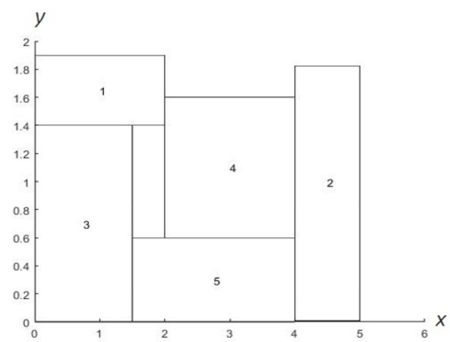


Figure 1: A non-guillotine pattern corresponding to 1.

Then, by the conditions (i)-(iv) we have

$$\left.\begin{array}{r} z_{3r} > z_{1b}, \\ z_{1b} > z_{4l}, \\ z_{4l} > z_{5t}, \\ z_{5t} > z_{3r} \end{array}\right\} \Rightarrow \; z_{3r} > z_{3r}. \tag{19}$$

The four inequalities in the left side of (19) follow from (ii), (iii), (i) and (iv), respectively. This contradiction ($z_{3r} > z_{3r}$) shows that the above non-guillotine pattern cannot provide any label sets that satisfy conditions (i)-(iv).

2.3 GSPP in the form of an MILP model

- Condition (i) in the form of mathematical modeling constraints:

We can present Condition (i) in the form of mathematical constraints by introducing binary variables δ^1_{ij} and γ^1_{ij} for $i \neq j$ and adding the following linear constraints to the problem (1)-(10):

$$x_i \leq x_j + \bar{x}(1-\delta^1_{ij}) \qquad i \neq j, \tag{20}$$

$$x_j + p_j s_j + q_j(1-s_j) - \bar{x}(1-\gamma^1_{ij}) \leq x_i \qquad i \neq j, \tag{21}$$

$$z_{il} + M(\delta^1_{ij} + \gamma^1_{ij}) \geq z_{jb} - M(1-d_{(ij)}) + 1 \qquad i \neq j, \tag{22}$$

$$z_{il} + M(\delta^1_{ij} + \gamma^1_{ij}) \geq z_{jt} - M(1-c_{(ij)}) + 1 \qquad i \neq j. \tag{23}$$

Note that according to (20), if $x_i \le x_j$ then δ^1_{ij} can take any of the values 0 or 1. Similarly, according to (21), if $x_i \ge x_j + p_j s_j + q_j(1-s_j)$ then γ^1_{ij} can take any of the values 0 or 1. Therefore, in such cases no additional restrictions are imposed to the problem (1)-(10) and none of its feasible solutions are excluded. But, when the top edge or the bottom edge of item j is an obstacle to the left edge of item i i.e. $x_j < x_i < x_j + p_j s_j + q_j(1-s_j)$, the constraints (20) and (21) force δ^1_{ij} and γ^1_{ij} to take zero.

On the other hand, (22) does not impose any restrictions and does not exclude any feasible solution of (1)-(10) unless $\delta^1_{ij} = \gamma^1_{ij} = 0$ and $d_{(ij)} = 1$. Similarly, constraint (23) does not impose any restriction unless $\delta^1_{ij} = \gamma^1_{ij} = 0$ and $c_{(ij)} = 1$.

To summarize, only when the if-clause of the condition (i) is satisfied, δ^1_{ij} and γ^1_{ij} are forced by (20) and (21) to take a value of zero and hence the obstacle edge (the bottom edge or the top edge) is forced by (22) or (23) to take a label smaller than z_{il}. In any other cases, (20)-(23) does not impose any additional restrictions and do not exclude any feasible solutions (patterns) of the problem (1)-(10).

The above discussion shows that the Condition (i) can be presented by (20)-(23). By similar discussions, one concludes that the Conditions (ii)-(iv) can also be presented in the form of linear mathematical programming constraints.

- Condition (ii) in the form of mathematical modeling constraints:

$$x_i + p_i s_i + q_i(1-s_i) \le x_j + \bar{x}(1-\delta^2_{ij}) \qquad i \ne j, \tag{24}$$

$$x_j + p_j s_j + q_j(1-s_j) - \bar{x}(1-\gamma^2_{ij}) \le x_i + p_i s_i + q_i(1-s_i) \qquad i \ne j, \tag{25}$$

$$z_{ir} + M(\delta^2_{ij} + \gamma^2_{ij}) \ge z_{jb} - M(1-d_{(ij)}) + 1 \qquad i \ne j, \tag{26}$$

$$z_{ir} + M(\delta^2_{ij} + \gamma^2_{ij}) \ge z_{jt} - M(1-c_{(ij)}) + 1 \qquad i \ne j. \tag{27}$$

- Condition (iii) in the form of mathematical modeling constraints:

$$y_i \le y_j + \bar{y}(1-\delta^3_{ij}) \qquad i \ne j, \tag{28}$$

$$y_j + q_j s_j + p_j(1-s_j) - \bar{y}(1-\gamma^3_{ij}) \le y_i \qquad i \ne j, \tag{29}$$

$$z_{ib} + M(\delta^3_{ij} + \gamma^3_{ij}) \ge z_{jl} - M(1-b_{(ij)}) + 1 \qquad i \ne j, \tag{30}$$

$$z_{ib} + M(\delta^3_{ij} + \gamma^3_{ij}) \ge z_{jr} - M(1-a_{(ij)}) + 1 \qquad i \ne j. \tag{31}$$

- Condition (iv) in the form of mathematical modeling constraints:

$$y_i + q_i s_i + p_i(1-s_i) \le y_j + \bar{y}(1-\delta^4_{ij}) \qquad i \ne j, \tag{32}$$

$$y_j + q_j s_j + p_j(1-s_j) - \bar{y}(1-\gamma^4_{ij}) \le y_i + q_i s_i + p_i(1-s_i) \qquad i \ne j, \tag{33}$$

$$z_{it} + M(\delta^4_{ij} + \gamma^4_{ij}) \ge z_{jl} - M(1-b_{(ij)}) + 1 \qquad i \ne j, \tag{34}$$

$$z_{it} + M(\delta^4_{ij} + \gamma^4_{ij}) \ge z_{jr} - M(1-a_{(ij)}) + 1 \qquad i \ne j. \tag{35}$$

Based on the above results and discussions, we can present the GSPP by the following MILP model:

$$Minimize \quad Y \tag{36}$$

$$s.t.$$

$$(GSPP) \qquad (2)-(10),\ (20)-(35), \tag{37}$$

$$z_{il}, z_{ir}, z_{it}, z_{ib} \ge 0 \qquad i \in I, \tag{38}$$

$$\delta^k_{ij}, \gamma^k_{ij} \in \{0,1\} \qquad k = 1,2,3,4,\ i \ne j. \tag{39}$$

It is not difficult to see that *GSPP* model has $6n+1$ continuous variables, $10n^2-9n$ binary variables and $\frac{37}{2}n^2-\frac{33}{2}n$ constraints. Hence, both the number of decision variables and the number of constraints are of order $O(n^2)$. It is worth mentioning that the number of variables and constraints of the mathematical model presented in [23] are about $\frac{3n^4}{4}$ and $2n^4$, respectively.

Example 2. Recall the rectangular items of Example 1. The non-guillotine pattern of Figure 1 is an optimal strip packing of the items into a strip of width 5. Note that the minimum used length of the strip is 1.9. The same items and strip were considered to find an optimal guillotine packing pattern by solving the related *GSPP* model. The resulting pattern is as presented in Figure 2. As we can see, the minimum length of the strip needed to pack the items is 2 (>1.9) since non-guillotine patterns are not accepted. Note that in *GSPP* unlike many heuristics, items are not necessarily packed at the bottom-left corners. For example, although item number 5 can be moved to the left without changing the used height of the strip, it is packed at the rightmost position.

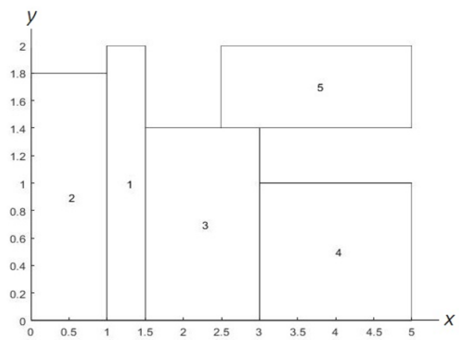


Figure 2: An optimal guillotine packing pattern corresponding to the items of Example 1.

2.4 Valid inequalities

It is worth noting that, for each pair of distinct items i and j in I, the variales a_{ij}, b_{ij}, c_{ij} and d_{ij} form a special ordered set of type 1 (SOS1). In other words, they form a set of variables that at most one of them can take a non-zero value (see (6) and (10)). Most of modern MIP solvers provide the ability to intoduce SOSs in optimization models. Indeed, we can eliminate Constraints (6) by introducing a_{ij}, b_{ij}, c_{ij} and d_{ij} as a SOS1. This reduces the number constraints of the *GSPP* model to $18n^2-16n$. The benefit of using SOSs is that the search procedure (in the search tree) will generally be noticeably faster. In fact, the number of binary variables at each node reduces to $8n^2-7n$. We take advantage of this feature in our implementation.

Now, assume that in a certain node of the search tree, a_{ij} is set to 1. Then, according to (2), item i is placed to the right of item j and according to (20) and (24), the variables δ^1_{ij} and δ^2_{ij} cannot take the value of 1. But, despite our desire, they may take any real value in the interval $[0,1)$ in the solution of the continuous relaxation of the problem at that node. Also, in this case, γ^1_{ij} and γ^2_{ij} are permitted to take any real value in the interval $[0,1]$ without imposing any restrictions to the problem (see (21)-(23) and (25)-(27)). Hence, it is worth to force these variables to take integer values in the solution of the continuous relaxation of the problem at each node. The following inequalities are a set of valid inequalities that help to achieve this goal:

$$4a_{ij} \le (1-\delta^1_{ij})+\gamma^1_{ij}+(1-\delta^2_{ij})+\gamma^2_{ij} \qquad i,j\in I, \quad i<j. \tag{40}$$

It is clear that, if $a_{ij}=1$ then δ^1_{ij} and δ^2_{ij} must take the value of 0, and γ^1_{ij} and γ^2_{ij} must take the value of 1.

By a similar discussion, we can show that (41)-(43) are also valid inequalities:

$$4b_{ij} \le \delta^1_{ij} + (1-\gamma^1_{ij}) + \delta^2_{ij} + (1-\gamma^2_{ij}) \qquad i,j \in I, \quad i<j, \tag{41}$$
$$4c_{ij} \le (1-\delta^3_{ij}) + \gamma^3_{ij} + (1-\delta^4_{ij}) + \gamma^4_{ij} \qquad i,j \in I, \quad i<j, \tag{42}$$
$$4d_{ij} \le \delta^3_{ij} + (1-\gamma^3_{ij}) + \delta^4_{ij} + (1-\gamma^4_{ij}) \qquad i,j \in I, \quad i<j. \tag{43}$$

Another benefit of using SOSs is that fixing a_{ij} or b_{ij} at 0 at a node simply leads to (20)-(27) becoming redundant and being detected and eliminated by the solver in the presolve phase at that node. Similarly, fixing c_{ij} or d_{ij} at 0 at a node leads to (28)-(35) becoming redundant and being detected and eliminated by the solver in the presolve phase at that node. Elimination of these redundant constraints, significantly reduces the size of the problems to be solved at each node.

3 Numerical experiments

In order to check the performance of the proposed model *GSPP*, we used a series of small and medium sized instances of some benchmark instances from the literature. As we mentioned before, we did not find (in the literature) any mathematical model or any algorithm capable to globally solve the guillotine strip packing problem except the MILP model proposed in [23]. But, the model of [23] is of order $O(n^4)$ and no experimental results have been reported for it. Therefore, we decided to compare our model with heuristic algorithms. More than 10 heuristic algorithms for the guillotine oriented strip packing problem are compared in [27]. We compare the MILP model *GSPP* presented by (36)-(39) with the best functioned heuristic for each instances. Since no execution time is reported for the mentioned heuristics, we consider some time limit (as shown in Table 1) and compare the best objective values obtained from our MILP model with the best one obtained from the heuristics. Furthermore, since an oriented version of the strip packing problem has been considered in [27], we also considered an oriented version of *GSPP* (i.e. we set $s_i = 1$ for all $i = 1,,n$) to be able to compare our results with theirs. The *GSPP* model was implemented using GAMS 24.2.2 and solved using CPLEX 12.6.0 solver on a laptop with Intel(R) Core(TM) i7-6498DU CPU @ 2.5GHz, 12 GB RAM, 64-bit.

3.1 Benchmark data

The set of benchmark instances that we applied to test efficiency of the MILP model *GSPP* (which are available and can be accessed via http://or.dei.unibo.it/library/2dpacklib) includes instances with up to 29 rectangular items instances from N, T series , instances with up to 29 items from C series , instances with up to 30 items from GCUT series and all the instances from NGCUT series .

3.2 Results

The summary of the achieved results is presented in Table 1. The column LB indicates a lower bound for the problems some of which are obtained from the simple formula $(1/W)\sum_{i=1}^{n} w_i h_i$ and some others are obtained from heuristics in the literature (see [21]). The column Best Heur. presents the best value obtained from the best functioned heuristics among more than 10 heuristics studied in [27]. The column GSPP presents the best values obtained by solving our MILP model *GSPP* within a time limit of T.L.. Finally, the column Best Time indicates the least execution time needed to obtain the values in column

GSPP. For example, the first row of the table (corresponding to 'T1' class) states that the *GSPP* model solution leads to an objective value of 223.2 after 355 seconds and no better objective value is obtained within 900 seconds. An asterisk is placed next to some values in the *GSPP* column to indicate that the obtained values are optimal.

Table 1: Comparison of GSPP with the best functioned heuristics in the literature.

Class	*Num.ofInst.*	*Num.ofItems*	*LB*	*BestHeur.*	*GSPP*	*T.L.(sec)*	*BestTime*
T1	5	17	200	274.4	223.2	900	355
T2	5	25	200	263.2	229.2	1800	1375
T3	5	29	200	251.8	238.8	3600	2735
N1	5	17	200	242.6	219.6	900	539
N2	5	25	200	244.8	235.4	1800	1129
N3	5	29	200	246.0	230.6	3600	2702
C1	3	16/17	20	22.0	20.7	900	407
C2	3	25	15	17.0	17.3	1800	1206
C3	3	28/29	30	35.3	35.3	3600	2150
GCUT1	1	10	1016	1016	1016*	900	0
GCUT2	1	20	1133	1347	1266	1800	1681
GCUT3	1	30	1803	1810	1872	3600	1264
NGCUT1	1	10	23	25	23*	900	2
NGCUT2	1	17	30	33	30*	900	16
NGCUT3	1	21	28	31	30	900	274
NGCUT4	1	7	20	23	20*	900	0
NGCUT5	1	14	36	37	36*	900	8
NGCUT6	1	15	31	35	31*	900	48
NGCUT7	1	8	20	20	20*	900	0
NGCUT8	1	13	33	38	35	900	70
NGCUT9	1	18	49	60	54	900	48
NGCUT10	1	13	80	85	80*	900	8
NGCUT11	1	15	52	63	57	900	12
NGCUT12	1	22	87	91	87*	900	62

Based on the number of items to be packed (difficulty of the instances), different time limits have been considered for instances of N, T, 'C' and 'GCUT' classes ($T.L. = 900, 1800, 3600$ for instances with less than 20 items, instances with 20 to 25 items and instances with more than 25 items, respectively). For all of instances in 'NGCUT' class (which are relatively simpler instances), a time limit of 900 seconds is considered.

The results show that the proposed model is able to find better solutions than the most known heuristics in almost all test instances. As can be seen, instances with $n \leq 10$ are solved optimally in just a few seconds. However, the solution time of the proposed *GSPP* increases greatly when the size of instances increases.

4 Conclusions

An $O(n^2)$ polynomial MILP relaxation of the *GSPP* (inspired by a *sequence sub-tour elimination* technique first proposed for the traveling salesman problem) was presented. Although the proposed model does not guarantee the achievement of optimal patterns, it can be applied to problems of relatively larger size compared to the only exact model in the literature [23]. Furthermore, although the efficiency of our relaxation model decreases significantly with increasing the size of the instances, numerical results show that the solutions obtained from our model are superior to several existing heuristic algorithms in the literature for not too large instances. Therefore, the author believes that better performances of the proposed model can be achieved if it can be combined with heuristics and decomposition techniques where several smaller problems are solved instead of solving the original problem once.

References

[1] R. Alvarez-Valdes, F. Parreno, J. M. Tamarit, *A tabu search algorithm for a two-dimensional non-guillotine cutting problem*, Eur. J. Oper. Res. **183** (2007) 1167–1182.

[2] B.S. Baker, E.G. Coffman Jr, R.L. Rivest, *Orthogonal packing in two dimensions*, SIAM J. Comput. **9** (1980) 846–855.

[3] M.A. Boschetti, A. Mingozzi, *The two-dimensional finite bin packing problem. Part I: New lower bounds for the oriented case*, 4OR **1** (2003) 27–42.

[4] M.A. Boschetti, A. Mingozzi, *The two-dimensional finite bin packing problem. Part II: New lower and upper bounds*, 4OR **1** (2003) 135–147.

[5] E.K. Burke, G. Kendall, G. Whitwell, *A new placement heuristic for the orthogonal stock-cutting problem*, Oper. Res. **52** (2004) 655–671.

[6] P.M. Castro, I.E. Grossmann, *From time representation in scheduling to the solution of strip packing problems*, Comput. Chem. Eng. **44** (2012) 45–57.

[7] P.M. Castro, J.F. Oliveira, *Scheduling inspired models for two-dimensional packing problems*, Eur. J. Oper. Res. **215** (2011) 45–56.

[8] C.S. Chen, S. Sarin, R. Balasubramanian, *A mixed-integer programming model for a class of assortment problems*, Eur. J. Oper. Res. **65** (1993) 362–367.

[9] H.I. Christensen, A. Khan, S. Pokutta, *Approximation and online algorithms for multidimensional bin packing: A survey*, Comput. Sci. Rev. **24** (2017) 63–79.

[10] F. Furini, E. Malaguti, *Models for the two-dimensional two-stage cutting stock problem with multiple stock size*, Comput. Oper. Res. **40** (2013) 1953–1962.

[11] F. Furini, E. Malaguti, R. M. Duran, *A column generation heuristic for the two-dimensional two-staged guillotine cutting stock problem with multiple stock size*, Eur. J. Oper. Res. **218** (2012) 251–260.

[12] F. Furini, E. Malaguti, D. Thomopulos, *Modeling two-dimensional guillotine cutting problems via integer programming*, INFORMS J. Comput. **28** (2016) 736–751.

[13] Y.H. Huang, H.C. Lu, Y.C. Wang, Y.F. Chang, C.K. Gao, *A global method for a two-dimensional cutting stock problem in the manufacturing industry*, in Application of Decision Science in Business and Management, IntechOpen, 2020.

[14] M. Iori, V.L. de Lima, S. Martello, F.K. Miyazawa, M. Monaci, *Exact solution techniques for two-dimensional cutting and packing*, Eur. J. Oper. Res. **289** (2021) 399–415.

[15] M. Iori, S. Martello, M. Monaci, *Metaheuristic Algorithms for the Strip Packing Problem*, US: Springer, 2003.

[16] T.W. Leung, C.K. Chan, M.D. Troutt, *Application of a mixed simulated annealing-genetic algorithm for a 2D packing problem*, Eur. J. Oper. Res. **145** (2003) 530–542.

[17] H.L. Li, C.T. Chang, *An approximately global optimization method for assortment problems*, Eur. J. Oper. Res. **105** (1998) 604–612.

[18] A. Lodi, S. Martello, D. Vigo, *Models and bounds for two-dimensional level packing problems*, J. Comb. Optim. **8** (2004) 363–379.

[19] A. Lodi, M. Monaci, *Integer linear programming models for 2-staged two-dimensional knapsack problems*, Math. Program. **94** (2003) 257–278.

[20] R. Macedo, C. Alves, J.M. Valerio de Carvalho, *Arc-flow model for the two-dimensional guillotine cutting stock problem,* Comput. Oper. Res. **37** (2010) 991–1001.

[21] S. Martello, M. Monaci, D. Vigo, *An exact approach to the strip-packing problem*, INFORMS J. Comput. **15** (2003) 310–319.

[22] M. Martin, R. Morabito, P. Munari, *A bottom-up packing approach for modeling the constrained two-dimensional guillotine placement problem*, Comput. Oper. Res. **115** (2020) 104851.

[23] S.B. Messaoud, C. Chu, M.L. Espinouse, *Characterization and modelling of guillotine constraints*, Eur. J. Oper. Res. **191** (2008) 112–126.

[24] C. Miller, A. Tucker, R. Zemlin, *Integer programming formulations and travelling salesman problems*, J. ACM **7** (1960) 326–329.

[25] M. Monaci, P. Toth, *A set-covering-based heuristic approach for bin-packing problems*, INFORMS J. Comput. **18** (2006) 71–85.

[26] M. Mrad, *An arc flow-based optimization approach for the two-stage guillotine strip cutting problem*, J. Oper. Res. Soc. **66** (2015) 1850–1859.

[27] N. Ntene, J. van Vuuren, *A survey and comparison of guillotine heuristics for the 2D oriented offline strip packing problem*, Discrete Optim. **6** (2009) 174–188.

[28] F. Parreno, R. Alvarez-Valdes, J.F. Oliveira, J.M. Tamarit, *A hybrid GRASP/VND algorithm for two- and three-dimensional bin packing*, Ann. Oper. Res. **179** (2010) 203–220.

[29] J. Puchinger, G.R. Raidl, *Models and algorithms for three-stage two-dimensional bin packing*, Eur. J. Oper. Res. **183** (2007) 1304–1327.

[30] C.D. Rodrigues, A.C. Cherri, S.A. de Araujo, *Strip based compact formulation for two-dimensional guillotine cutting problems*, Comput. Oper. Res. **149** (2023) 106044.

[31] S.M. Shperling, Y.A. Kochetov, *A knapsack problem for rectangles under center-of-gravity constraints*, J. Appl. Ind. Math. **16** (2022) 563–571.

[32] E. Silva, F. Alvelos, J.M. Valerio de Carvalho, *An integer programming model for two- and three-stage two-dimensional cutting stock problems*, Eur. J. Oper. Res. **205** (2010) 699–708.

[33] L. Wei, H. Qin, B. Cheang, X. Xu, *An efficient intelligent search algorithm for the two-dimensional rectangular strip packing problem*, Int. Trans. Oper. Res. **23** (2016) 65–92.