LIII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2021

Enhanced solution for the irregular strip packing problem: valid inequalities and branching priorities

Aline A. S. Leão
Departamento de Matemática - Universidade Estadual de Londrina
Rod. Celso Garcia Cid, PR-445, km 380, 86057-970, Londrina-PR, Brasil
aasleao@uel.br

Franklina M. B. Toledo
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo
Av. Trabalhador São-carlense, 13566-590, São Carlos - SP, Brasil
fran@icmc.usp.br

RESUMO

Problemas de corte e empacotamento em faixas com peças irregulares consistem em empacotar um conjunto de peças regulares e irregulares em uma placa com altura fixa e comprimento variável. Todas as peças são empacotadas de modo que não se sobreponham e o comprimento utilizado da placa seja minimizado. Alguns modelos matemáticos foram propostos recentemente na literatura, porém resolver estes problemas na otimalidade ainda é um desafio e somente para algumas instâncias com poucas peças é possível encontrar a solução ótima. Neste trabalho, aplicamos uma nova estratégia de ramificação no método *branch-and-bound* e adicionamos desigualdades válidas em um modelo da literatura. O uso dessas estratégias melhorou o desempenho do *solver* comercial utilizado para resolver o modelo de programação inteira mista, sendo possível determinar a solução ótima de mais 10 instâncias não encontradas anteriormente.

PALAVRAS CHAVE. Problemas de corte e empacotamento; peças irregulares; programação inteira mista

PM - Programação Matemática; POI - PO na Indústria

ABSTRACT

The irregular strip packing problem consists of packing a set of regular and irregular pieces on a board with fixed width and infinite length. All pieces are packed in such a way that they do not overlap while minimizing the used length of the board. Despite a few mathematical models having been proposed in the literature, solving the problem to optimality is still a challenge, and for only very few small instances, it is possible to find an optimal solution. In this paper, we apply a different branching strategy in the branch-and-bound method and add valid inequalities in a mathematical model of the literature. The usage of branching strategies and valid inequalities improve the performance of the commercial solver used to solve the mixed-integer programming model and provide the optimal solution for up to ten instances not proved in the previous paper.

KEYWORDS. Cutting and packing problems; irregular pieces; mixed-integer programming.

PM - Programação Matemática; POI - PO na Indústria

1. Introduction

Strip packing problems consist of packing a set of pieces on a board with fixed width and infinite length in a way that the pieces do not overlap and the used length is minimized. In the irregular version of the problem, at least one piece must be irregularly shaped. This problem has many industrial applications such as leather, garment, furniture and sheet metal industries and has been extensively studied in the literature. According to the typology proposed by Wäscher et al. [2007], the irregular strip packing problem can be categorized as Open Dimension Problems.

The main difficulty in solving irregular packing problems is related to geometry. For the development of heuristics, exact methods and mathematical models, it is necessary to implement geometric tools to represent the problem. Bennell and Oliveira [2008] review some tools that can be used to ensure that pieces do not overlap each other and are placed entirely inside of the board. Raster points methods, direct trigonometry, no-fit polygon, inner-fit polygon and phi-functions allow the computational and mathematical representation of pieces and their positioning on the board.

Many solution approaches have been proposed in the literature to solve the irregular strip packing problem. The use of heuristics has been applied to solve large problems and obtain compact layouts, which is important to solve practical applications. The bottom-left heuristic has been extensively applied to obtain an initial solution. Then, more complex methods are applied to improve the initial solution. A variety of heuristics, metaheuristics and compact and separation models are used to obtain better layouts. As far as the authors know, the better results were obtained by the heuristics of Sato et al. [2019], Elkeran [2013], Sato et al. [2012], Leung et al. [2012], Imamichi et al. [2009] and Egeblad et al. [2007].

A few mathematical models were proposed in the literature, which can be used to prove the optimality of only a few small problems. Mixed-integer linear programming models are discussed in Scheithauer and Terno [1993], Daniels et al. [1994], Dean [2002], Fischetti and Luzzi [2009], Alvarez-Valdes et al. [2013] and Cherri et al. [2016]. In these models, the pieces are represented by polygons where a reference point is defined for each piece, and they allow the continuous positioning of the reference point of the pieces on the board (continuous positioning models). Toledo et al. [2013] proposed an integer linear programming model that considers a discrete positioning of the pieces on the board by placing a grid on it (discrete positioning model), which was later improved by Rodrigues and Toledo [2017]. Leao et al. [2016] proposed a mixed-integer linear programming model that keeps the x-axis continuous while discretizes the y-axis (semi-continuous model). Then, the reference point of the pieces can be placed only in the stripes defined by the discretization of the y-axis. All the mixed-integer linear programming models use the no-fit polygon concept to avoid the overlap of pieces, but Scheithauer and Terno [1993] and Cherri et al. [2016] also proposed mathematical models based on direct trigonometry. A review of integer linear programming, mixed-integer linear programming, non-linear programming and constraint programming models is presented in Leao et al. [2020].

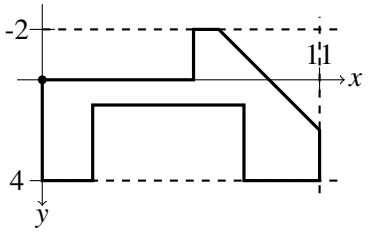
In this paper, we apply a set of valid inequalities to the model proposed in Leao et al. [2016]. In addition, different branching strategies were implemented in the branch-and-bound method. Fischetti and Luzzi [2009] and Alvarez-Valdes et al. [2013] showed that assigning branching priority to integer variables can improve the performance of commercial solvers. The addition of branching priorities also shows very powerful to the model of Leao et al. [2016], obtaining a better solution for most analyzed instances. By adding valid inequalities and branching priorities, the solver was able to improve different solutions in comparison with just the addition of branching priorities.

The remainder of this paper is organized as follows. In order to this paper be self-

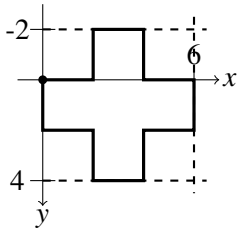
contained, Section 2 presents the mathematical model proposed in Leao et al. [2016]. In Sections 3 and 4, we describe the branching priorities and the valid inequalities, respectively. Computational results are discussed in Section 5. Finally, in Section 6, we draw some conclusions.

2. Problem definition

In this section, we review the mathematical model proposed in Leao et al. [2016], called semi-continuous model. The mathematical model can be stated as follows. A set of pieces must be placed on a board of fixed width (N) and infinite length, where the board is discretized in the y-axis. Each piece has a reference point that must be placed only in the stripes of the board. By discretizing the board, we reduce the solution space in comparison with the continuous positioning models. However, for a few instances, the quality of the packing layout can be inferior to the continuous positioning models. For the definition of the semi-continuous model, consider P as the number of pieces to be placed on the board. For each piece, let $r_i^{\min}$ be the minimum x-coordinate, $r_i^{\max}$ be the maximum x-coordinate, $t_i^{\min}$ be the minimum y stripe, and $t_i^{\max}$ be the maximum y stripe of the piece relative to its reference point. These parameters are used to define the inner fit rectangle in the x-coordinate ($r_i^{\min}$ and $r_i^{\max}$) and in the y-coordinate ($t_i^{\min}$ and $t_i^{\max}$). Figure 1 illustrates two pieces i and j, where the origin of the Cartesian coordinate system is placed at the reference point. The minimum and maximum x/y-coordinates and their values are also highlighted.



Piece i: $t^{min} = -2; t^{max} = 4;$
$r^{min} = 0; r^{max} = 11.$



Piece j: $t^{min} = -2; t^{max} = 4;$
$r^{min} = 0; r^{max} = 6.$

Figure 1: Representation of pieces.

To avoid the non-overlapping between each pair of pieces i and j, we use the no-fit polygon (NFP_{ij}). The no-fit polygon can be defined as the set of points that the reference point of piece j can be placed to be in contact with piece i. This set of points and the region outside the no-fit polygon define the feasible region where the reference point of piece j can be placed in order not to overlap piece i. In the semi-continuous model, the no-fit polygon is also discretized, where $n_{ij}^{\max}$ is the maximum stripe and $n_{ij}^{\min}$ is the minimum stripe. Consider a_{ij}^{kc} as the x-coordinate of the left end of stripe k related to concavity c, and b_{ij}^{kc} as the x-coordinate of the right end of stripe k related to concavity c of NFP_{ij}. Figure 2 shows the no-fit polygon between pieces i and j in Figure 1, where piece i is placed at the origin of the coordinate system and piece j can be placed only in the stripes outside or in the intersection between the stripes and the edges of the no-fit polygon. Few coordinates a_{ij}^{kc} and b_{ij}^{kc} are also represented. Note that for each stripe from -5 to 2 there is only one concavity, while for stripes 3 to 5 there are two concavities. For example, in stripe 3 we have a concavity associated with points a_{ij}^{31} and b_{ij}^{31}, and the second one is defined by the points a_{ij}^{32} and b_{ij}^{32}. Then, if the reference point of piece j is placed in stripe 3, it must be or at these points, or on the left-hand side of a_{ij}^{31}, or on the right-hand side of b_{ij}^{31} and on the left-hand side of a_{ij}^{32}, or on the right-hand side of b_{ij}^{32}. However, in this case point b_{ij}^{31} coincides with a_{ij}^{32}.

The continuous variable x_i defines the position of piece i in the x-coordinate, while variable $\delta_i^{s_i}$ defines its discrete position in the y-coordinate. $\delta_i^{s_i}$ is equal to 1 if the reference point of piece i is in stripe s_i of the board, and 0 otherwise, $1 \leq i \leq P$, $0 \leq s_i \leq N-1$. A set of auxiliary variables is necessary to determine the position of each piece in reference to another one, denoted by γ_{ij}^c. This variable is equal to 1 if the reference point of piece i is on the right of piece j in concavity c or 0 if it is on the left, $1 \leq i < j \leq P$, $c = 1, ..., C_{ij}^{s_j - s_i}$, where the reference point of pieces i and j are in stripes s_i and s_j, respectively. $C_{ij}^{s_j - s_i}$ denotes the total number of concavities for $s_j - s_i$. Finally, variable z is the length of the board, that must be minimized.

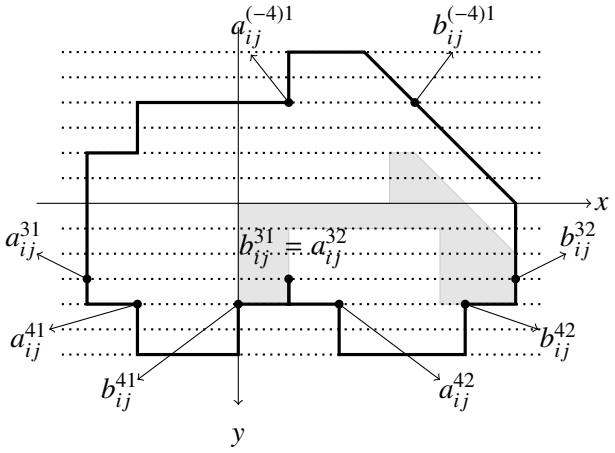


Figure 2: No-fit polygon between pieces i and j (NFP_{ij}).

The model described in Leao et al. [2016] is given as follows.

$$\text{minimize} \quad z, \tag{1}$$

$$\text{subject to:}$$

$$x_i + r_i^{max} \leq z, \qquad 1 \leq i \leq P, \tag{2}$$

$$x_i \leq x_j - b_{ij}^{(s_j - s_i)c} + \gamma_{ij}^c M + (1 - \delta_i^{s_i})M + (1 - \delta_j^{s_j})M, \tag{3}$$
$$1 \leq i \leq P, \quad i+1 \leq j \leq P, \quad c = 1, ..., C_{ij}^{s_j - s_i},$$
$$-t_i^{min} \leq s_i \leq N - t_i^{max}, \quad -t_j^{min} \leq s_j \leq N - t_j^{max}, \quad n_{ij}^{min} \leq s_j - s_i \leq n_{ij}^{max},$$

$$x_i \geq x_j - a_{ij}^{(s_j - s_i)c} - (1 - \gamma_{ij}^c)M - (1 - \delta_i^{s_i})M - (1 - \delta_j^{s_j})M, \tag{4}$$
$$1 \leq i \leq P, \quad i+1 \leq j \leq P, \quad c = 1, ..., C_{ij}^{s_j - s_i},$$
$$-t_i^{min} \leq s_i \leq N - t_i^{max}, \quad -t_j^{min} \leq s_j \leq N - t_j^{max}, \quad n_{ij}^{min} \leq s_j - s_i \leq n_{ij}^{max},$$

$$\sum_{s=-t_i^{min}}^{N-t_i^{max}} \delta_i^s = 1, \qquad 1 \leq i \leq P, \tag{5}$$

$$x_i \geq -r_i^{min}, \qquad 1 \leq i \leq P, \tag{6}$$

$$\delta_i^{s_i} \in \{0,1\}, \qquad 1 \leq i \leq P, \quad -t_i^{min} \leq s_i \leq N - t_i^{max}, \tag{7}$$

$$\gamma_{ij}^c \in \{0,1\}, \qquad 1 \leq i < j \leq P, \quad c = 1, ..., C_{ij}^{s_j - s_i},$$
$$-t_i^{min} \leq s_i \leq N - t_i^{max}, \quad -t_j^{min} \leq s_j \leq N - t_j^{max}. \tag{8}$$

The objective function (1) minimizes the length of the board used to pack all pieces that is defined in constraints (2). Constraints (3) and (4) are the non-overlapping constraints between pieces i and j. These constraints are active only if the relative position between both pieces are between n_{ij}^{min} and n_{ij}^{max} and if piece j is on the left (constraints (3)) or on the right (constraints(4)) of piece i. Constraints (5) ensure the vertical containment of the pieces, where the reference point of the pieces must be in a stripe between $-t_i^{min}$ and $N - t_i^{max}$, while constraints (6) ensure the horizontal containment. The domain of the variables is defined in constraints (7) and (8).

3. Branching priorities

Some commercial solvers allow us to define different branching strategies for integer and binary variables in mathematical models by assigning branching priorities. This procedure can help to guide the solver in the exploration of the branch-and-bound tree. For irregular strip packing problems, this has been used in Fischetti and Luzzi [2009] and Alvarez-Valdes et al. [2013]. Fischetti and Luzzi [2009] describe an algorithm that defines a clique of consistent fixings by assigning priorities to the non-overlapping variables. After choosing an initial piece, in each iteration of the algorithm, another piece is selected and a weight is assigned to the non-overlapping variables between the current piece and the previous ones. However, the authors did not specify an order to select the pieces. Then, Alvarez-Valdes et al. [2013] investigate four orderings: original Fischetti and Luzzi's priorities (no initial ordering of pieces); pieces ordered by non-increasing length; pieces ordered by non-increasing area; pieces ordered by non-increasing area of the slices defined by the no-fit polygon. Their computational experiments showed that ordering the pieces by area reached more optimal solutions within the same time limit.

We propose a branching strategy that is suitable for the semi-continuous model. The procedure implemented defines the priority according to the area of the pieces. There are two sets of variables to define the priorities: $\delta_i^{s_i}$ and γ_{ij}^c. The binary variable $\delta_i^{s_i}$ is used to define the position of piece i on the board, which is equal to 1 if the reference point of piece i is in strip s_i. Variable γ_{ij}^c is used to define the relative position between pieces i and j in a given concavity c, which is equal to 1 if piece i is on the right of piece j, and 0 if piece i is on the left of piece j. By denoting $area(i)$ as the area of piece i and $pri(variable)$ as the priority of the variable, the procedure for each variable set can be given by equations (9) and (10).

$$pri(\gamma_{ij}^c) = \alpha * \max\{area(i), area(j)\} + \beta * \min\{area(i), area(j)\}, \tag{9}$$

where $\alpha + \beta = 1$. In this priority set, we consider the convex combination of the area of two pieces. Note that if pieces i and j have the same area, the priority assigned to the variable is equal to their area. In addition, if the no-fit polygons have two or more concavities, the solver chooses which variable must be fixed first.

$$pri(\delta_i^{s_i}) = area(i) + s_i. \tag{10}$$

The priority set given in (10) is based on the bottom-left heuristic, where we try first to place the pieces at the bottom of the board.

4. Valid inequalities

In this section, we apply a set of valid inequalities that is based on a valid inequality proposed in Rodrigues and Toledo [2017]. In the semi-continuous model, each piece is unique, we do not deal with types of pieces. However, if we group pieces of the same type, constraints (2) can be summed up for each piece type. For this, let $\bar{n}$ be the total number of piece types, $\mathcal{P}_j$ be the set

of piece indices of type j, and $|\mathcal{P}_j|$ be cardinality of $\mathcal{P}_j$, $j = 1, ..., \bar{n}$. Then, by lifting the coefficients of constraints (2), the valid inequality for each piece type is given by the following constraint:

$$\sum_{i \in \mathcal{P}_j} (x_i + r_i^{max}) \leq |\mathcal{P}_j| * z, 1 \leq j \leq \bar{n}. \tag{11}$$

These valid inequalities provide a lower bound for the board length and consequently for objective function (1). Moreover, they can be useful to improve the performance of branching methods.

5. Computational results

Computational experiments were performed on an Intel Core i7 at 3.4 GHz and 16 GB of RAM. The mathematical models were implemented in C/C++ language and solved by CPLEX 12.6. We have carried out computational experiments using 72 instances from the literature. The first set of instances is grouped in Three, Dighe, Poly1a, Fu, Jakobs1 and Jakobs2. Some variations for Fu, Jakobs1 and Jakobs2 are considered by selecting a subset of pieces from the original instances. Fun is composed of the first n pieces. J1-b-c-d and J2-b-c-d are composed of b pieces randomly selected, where the board has width c, and d represents the instance number. The second set of instances is composed of 34 instances that can be grouped in Blazewicz (Blaze), RCO and Shapes. The number n in Blaze$n$ and RCOn represents the number of piece repetitions. In addition, RCO is obtained by considering the convex hull of the seven pieces in Blazewicz. In BlazewiczPi_n and BlazewiczPiPj_$n_i n_j$, i and j represent the piece types that are repeated n_i and n_j times, respectively. Further details about the instances can be found in Alvarez-Valdes et al. [2013] and Toledo et al. [2013]. The proposed strategies are compared with the results of Leao et al. [2016] for the semi-continuous model that was run on the same computer previously described. In the computational tests, we consider a unitary distance between the stripes. The distance between the stripes influences the solution quality, where smaller distances allow more compact layouts for few instances, but it may be harder to solve the mathematical model.

In order to analyze the impact of the branching strategy developed and the influence of the valid inequalities in the model (1) - (8), we run two computational experiments. In the first one, we set the branching strategies (9) and (10) to CPLEX. The second one consists of the definition of the branching strategies and valid inequalities. A limit of 3600s was enforced to CPLEX for each instance. For the branching priority, we set different values to α and β in equation (9), and the best results were obtained for $\alpha = 0.8$ and $\beta = 0.2$.

Tables 1 and 2 show the results for instances considered in Alvarez-Valdes et al. [2013] and Toledo et al. [2013], respectively. We present the solution value and the computational time (in seconds) for each instance. TL means that the method reached the time limit without proving the optimality for the instance. Columns 3 and 4 show the results presented in Leao et al. [2016], columns 5 and 6 show the results of the branching strategy. The last two columns (7 and 8) show the results of adding branching strategy and valid inequalities to CPLEX.

Table 1 shows that by setting the branching priorities to the branch-and-cut method, the computational results improve. In addition, it obtains better results than branching priorities and the addition of the valid inequalities. Both new strategies prove optimality for most instances in Table 1. Moreover, the use of branching priorities and the addition of the valid inequalities prove optimality for nine and seven instances more than Leao et al. [2016], respectively. In particular, by considering the branching strategy, CPLEX was able to find the optimal solution for instance dighe2, in which there is no space among pieces. The computational time also decreases for instances that the optimality has been proved in the previous paper.

Table 1: Mathematical model results.

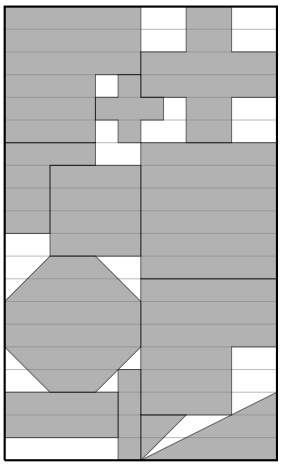
		Leao et al. [2016]		Branching strategy		Branching strategy & valid inequalities	
Instances	Pieces	Solution Value	Time(s)	Solution Value	Time(s)	Solution Value	Time(s)
three	3	6.00	0.31	**6.00**	**0.24**	6.00	0.54
threep2	6	9.67	0.98	9.67	0.75	**9.67**	**0.50**
threep2w9	6	8.00	3.10	**8.00**	**2.70**	8.00	2.82
threep3	9	14.00	852.69	14.00	179.99	**14.00**	**140.20**
threep3w9	9	11.33	TL	11.33	786.45	**11.33**	**590.19**
shapes4	4	24.00	0.60	24.00	0.64	**24.00**	**0.39**
shapes8	8	**26.00**	**48.96**	26.00	75.22	26.00	58.09
fu5	5	17.89	76.33	**17.89**	**15.81**	17.89	16.35
fu6	6	23.00	442.86	23.00	110.48	**23.00**	**91.24**
fu7	7	24.00	TL	**24.00**	**392.80**	24.00	532.01
fu8	8	24.00	TL	**24.00**	**906.51**	24.00	TL
fu9	9	**25.00**	**TL**	26.44	TL	26.00	TL
fu10	10	30.00	TL	30.00	TL	30.00	TL
fu	12	34.44	TL	**34.00**	**TL**	**34.00**	**TL**
dighe1	16	140.29	TL	135.42	TL	**132.33**	**TL**
dighe2	10	130.75	TL	**100.00**	**2057.03**	100.00	TL
poly1a	15	16.67	TL	**16.00**	**TL**	16.14	TL
J1-10-20-0	10	18.00	17.97	18.00	3.36	**18.00**	**3.24**
J1-10-20-1	10	17.00	26.98	17.00	4.31	**17.00**	**3.71**
J1-10-20-2	10	20.00	48.40	**20.00**	**4.53**	20.00	5.02
J1-10-20-3	10	20.75	110.99	**20.75**	**27.04**	20.75	51.46
J1-10-20-4	10	13.00	70.82	**13.00**	**11.60**	13.00	11.98
J1-12-20-0	12	12.00	TL	**12.00**	**383.70**	12.00	933.86
J1-12-20-1	12	10.00	TL	**10.00**	**169.72**	10.00	189.99
J1-12-20-2	12	12.00	TL	12.00	1387.20	**12.00**	**881.64**
J1-12-20-3	12	8.00	TL	**8.00**	**154.56**	8.00	312.91
J1-12-20-4	12	14.00	TL	13.00	TL	**13.00**	**961.22**
J1-14-20-0	14	14.00	TL	**12.00**	**1630.21**	13.00	TL
J1-14-20-1	14	12.00	TL	12.00	TL	12.00	TL
J1-14-20-2	14	**14.00**	**TL**	**14.00**	**TL**	14.25	TL
J1-14-20-3	14	11.00	TL	11.00	TL	11.00	TL
J1-14-20-4	14	14.00	TL	14.00	TL	14.00	TL
J2-10-35-0	10	24.25	TL	25.00	TL	**24.00**	**TL**
J2-10-35-3	10	20.75	TL	20.75	TL	20.75	TL
J2-10-35-4	10	20.00	TL	20.00	TL	20.00	TL
J2-12-35-1	12	**25.25**	**TL**	25.75	TL	26.00	TL
J2-12-35-3	12	22.80	TL	**22.00**	**TL**	23.00	TL
J2-14-35-1	14	30.50	TL	**30.00**	**TL**	**30.00**	**TL**

Table 2: Mathematical model results (continued).

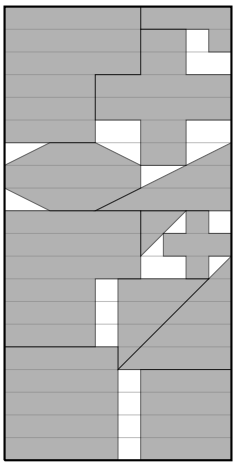
Instances	Pieces	Leao et al. [2016]		Branching strategy		Branching strategy & valid inequalities	
		Solution Value	Time(s)	Solution Value	Time(s)	Solution Value	Time(s)
Blaze1	7	7.50	4.92	**7.50**	**4.18**	7.50	4.32
Blaze2	14	**13.83**	**TL**	14.00	TL	**13.83**	**TL**
Blaze3	21	**20.67**	**TL**	21.00	TL	21.00	TL
Blaze4	28	29.00	TL	28.33	TL	**27.83**	**TL**
Blaze5	35	37.67	TL	**35.33**	**TL**	35.67	TL
RCO1	7	8.00	5.76	**8.00**	**4.53**	8.00	5.30
RCO1	14	15.00	TL	15.00	TL	15.00	TL
RCO3	21	**23.00**	**TL**	**23.00**	**TL**	23.67	TL
RCO4	28	32.33	TL	31.00	TL	**30.00**	**TL**
RCO5	35	**38.00**	**TL**	41.00	TL	38.33	TL
Shapes2	8	**14.00**	**6.61**	14.00	7.49	14.00	32.28
Shapes4	16	28.00	TL	28.00	TL	28.00	TL
Shapes5	20	**34.00**	**TL**	36.00	TL	35.00	TL
Shapes7	28	52.00	TL	52.00	TL	**50.00**	**TL**
Shapes9	34	**57.00**	**TL**	61.00	TL	59.00	TL
Shapes15	43	**78.00**	**TL**	85.00	TL	83.00	TL
BlazeP2_7	7	11.00	803.37	**11.00**	**196.16**	11.00	209.05
BlazeP2_14	14	18.00	TL	18.00	TL	18.00	TL
BlazeP2_21	21	25.00	TL	25.00	TL	25.00	TL
BlazeP2_28	38	35.50	TL	35.50	TL	35.50	TL
BlazeP2_35	35	42.50	TL	42.50	TL	42.50	TL
BlazeP4_7	7	10.00	TL	10.00	289.01	**10.00**	**257.08**
BlazeP4_14	14	18.33	TL	18.33	TL	18.33	TL
BlazeP4_21	21	27.00	TL	27.00	TL	27.00	TL
BlazeP4_28	28	36.50	TL	36.33	TL	**35.67**	**TL**
BlazeP4_35	35	46.00	TL	46.33	TL	**45.33**	**TL**
BlazeP2P4_4_3	7	10.50	1086.85	**10.50**	**38.52**	10.50	56.58
BlazeP2P4_7_7	14	17.67	TL	17.67	TL	17.67	TL
BlazeP2P4_11_10	21	26.50	TL	26.50	TL	26.50	TL
BlazeP2P4_14_14	28	**34.67**	**TL**	35.67	TL	35.33	TL
BlazeP2P4_18_17	35	43.83	TL	43.33	TL	43.33	TL
BlazeP2P4_21_21	42	53.33	TL	53.00	TL	**52.00**	**TL**
BlazeP2P4_28_28	56	71.00	TL	**70.50**	**TL**	91.00	TL
BlazeP2P4_35_35	70	**94.17**	**TL**	103.17	TL	109.50	TL

The instances considered in Table 2 are larger than those in Table 1. All methodologies are able to prove the optimality for instances with up to eight pieces. Branching priorities strategy and branching priorities with valid inequalities strategy proved the optimality for one instance that was not proved in Leao et al. [2016] (BlazeP4_7). For the instances that the three methods proved the optimality, the computational time of the new strategies was smaller, except for Shapes2. However, for large instances the solution value obtained in Leao et al. [2016] and by branching strategy with valid inequalities were better, where the first one found a better solution for nine instances and the last one for seven instances. The new optimal layouts under unitary y-axis discretization that CPLEX was able to prove the optimality considering the proposed strategies are shown in Figures

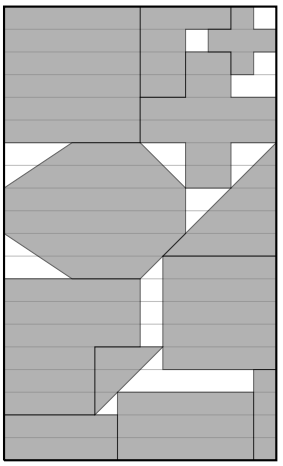
3 and 4.



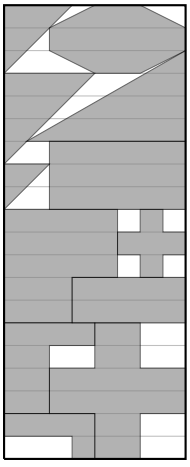
J1-12-20-0: Width: 20, z = 12.



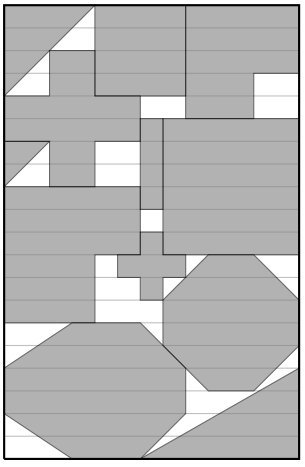
J1-12-20-1: Width = 20, z = 10.



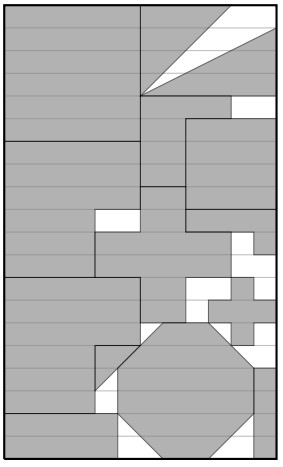
J1-12-20-2: Width: 20, z = 12.



J1-12-20-3: Width: 20, z = 8.



J1-12-20-4: Width = 20, z = 13.



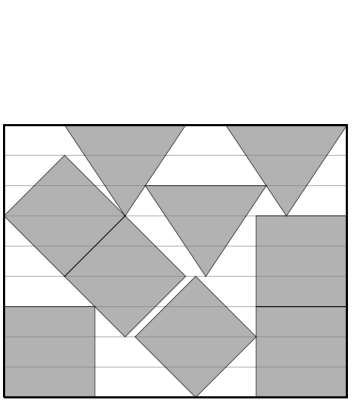
J1-14-20-0: Width = 20, z = 12.

Figure 3: Optimal layouts.

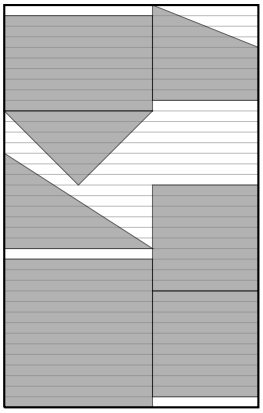
Table 3 gives an overview of the performance of the three methods. It shows the number of optimal solutions obtained by each method. As we can see, the strategy that considers only the branching priority outperforms the model solved by CPLEX in the default configuration and the addition of valid inequalities in the model combined with the branching strategy.

Table 3: The number of optimal solutions.

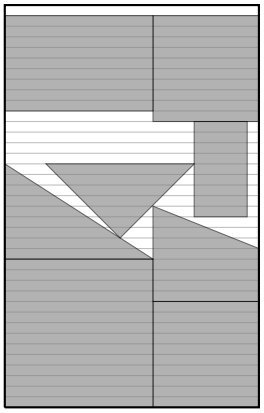
Leao et al. [2016]	Branching priorities	Branching Priorities & valid inequalities
18	28	26



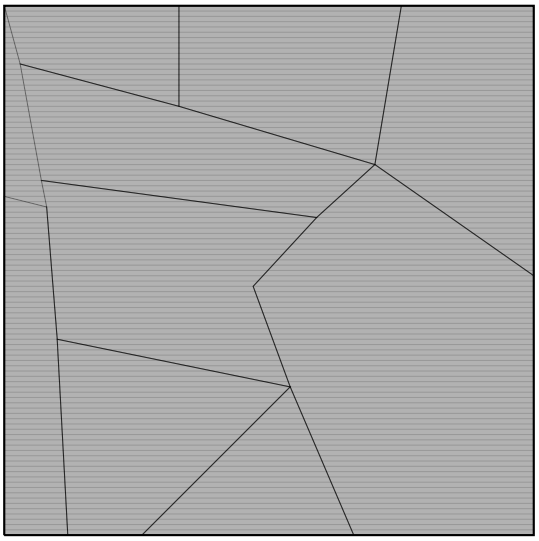
threep3w9: Width: 9, z = 11.33.



fu7: Width = 38, z = 24.



fu8: Width = 38, z = 24.



dighe2: Width: 100, z = 100.

Figure 4: Optimal layouts (Figure 3 continued).

We can conclude that the proposed strategies provide better solution values for instances with up to 14 pieces and can prove the optimality for more instances. In addition, their computational time is smaller than when CPLEX runs the model in the default configuration. If we consider only large instances, the branching strategies combined with valid inequalities are competitive with the model (1) - (8) solved by CPLEX in the default configuration.

6. Conclusions

In this paper, we apply some valid inequalities and branching strategies for the semi-continuous model proposed in Leao et al. [2016]. Better solutions are obtained for the semi-continuous model by applying branching strategies in the CPLEX's branch-and-bound method. Meanwhile, the valid inequalities improve the solution value of a few large instances. A future research direction could be to adapt the semi-continuous model to allow a discrete subset of piece rotations.

Acknowledgments

The authors would like to thank the anonymous referee for his/her comments. This research was supported by FAPESP CEPID (2013/07375-0), FAPESP (2012/21176-7) and CNPq

(308761/2018-9).

References

Alvarez-Valdes, R., Martinez, A., and Tamarit, J. (2013). A branch and bound algorithm for cutting and packing irregularly shaped pieces. *International Journal of Production Economics*, 145(2): 463–477.

Bennell, J. A. and Oliveira, J. F. (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397 – 415.

Cherri, L. H., Mundim, L. R., Andretta, M., Toledo, F. M., Oliveira, J. F., and Carravilla, M. A. (2016). Robust mixed-integer linear programming models for the irregular strip packing problem. *European Journal of Operational Research*, 253(3):570 – 583.

Daniels, K., Li, Z., and Milenkovic, V. J. (1994). Multiple containment methods. Technical report, Center for Research in Computing Technology, Division of Applied Sciences, Harvard University.

Dean, H. T. (2002). *Minimizing Waste in the 2-Dimensional Cutting Stock Problem*. PhD thesis, Department of Mechanical Engineering, University of Canterbury, Christchurch New Zealand.

Egeblad, J., Nielsen, B. K., and Odgaard, A. (2007). Fast neighborhood search for two- and three-dimensional nesting problems. *European Journal of Operational Research*, 183(3):1249 – 1266.

Elkeran, A. (2013). A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. *European Journal of Operational Research*, 231(3):757 – 769.

Fischetti, M. and Luzzi, I. (2009). Mixed-integer programming models for nesting problems. *Journal of Heuristics*, 15(3):201–226.

Imamichi, T., Yagiura, M., and Nagamochi, H. (2009). An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. *Discrete Optimization*, 6(4):345 – 361.

Leao, A. A. S., Toledo, F. M. B., Oliveira, J. F., and Carravilla, M. A. (2016). A semi-continuous mip model for the irregular strip packing problem. *International Journal of Production Research*, 54(3):712–721.

Leao, A. A. S., Toledo, F. M. B, Oliveira, J. F., Carravilla, M. A., and Alvarez-Valdés, R. (2020). Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, 282(3):803 – 822.

Leung, S. C., Lin, Y., and Zhang, D. (2012). Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem. *Computers and Operations Research*, 39(3):678 – 686.

Rodrigues, M. O. and Toledo, F. M. B. (2017). A clique covering mip model for the irregular strip packing problem. *Computers & Operations Research*, 87:221–234.

Sato, A. K., Martins, T. C., and Tsuzuki, M. S. G. (2012). An algorithm for the strip packing problem using collision free region and exact fitting placement. *Computer-Aided Design*, 44(8): 766 – 777.

Sato, A. K., Martins, T. C., Gomes, A. M., and Tsuzuki, M. S. G. (2019). Raster penetration map applied to the irregular packing problem. *European Journal of Operational Research*, 279(2): 657–671.

Scheithauer, G. and Terno, J. (1993). Modeling of packing problems. *Optimization*, 28(1):63–84.

Toledo, F. M. B., Carravilla, M. A., Ribeiro, C., Oliveira, J. F., and Gomes, A. M. (2013). The dotted-board model: A new MIP model for the nesting problem. *International Journal of Production Economics*, 145(2):478 – 487.

Wäscher, G., Haußner, H., and Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109 – 1130.