Improved Sets of Points for the Bin Packing Problem

Vinícius Loti de Lima
IC - UNICAMP
Av. Albert Einstein, 1251, Campinas/SP - 13083-852
vloti@ic.unicamp.br

Thiago Alves de Queiroz
IMTec - Universidade Federal de Goiás
Av. Dr. Lamartine Pinto de Avelar 1120, Catalão/GO, - 75704-020
taq@ufg.br

Manuel Iori
DISMI - University of Modena and Reggio Emilia
Via Giovanni Amendola, 2, Reggio Emilia, Italia - 42122
manuel.iori@unimore.it

Flávio Keidi Miyazawa
IC - UNICAMP
Av. Albert Einstein, 1251, Campinas/SP - 13083-852
fkm@ic.unicamp.br

RESUMO

O problema do empacotamento em bins possui um grande número de aplicações. Alguns de seus métodos de solução são baseados no posicionamento de itens, como formulações de fluxo em arcos. Abordagens para reduzir o número de posições em que os itens podem ser empacotados foram propostas desde os anos setenta até recentemente com o *meet-in-the-middle patterns*. Neste trabalho, propõe-se uma abordagem, baseada em um ordenamento arbitrário de itens, resultando em um conjunto de pontos melhorado. A ideia é eliminar posições redundantes através do alinhamento à direita e à esquerda de itens em soluções. Os experimentos computacionais sobre instâncias padrões mostraram que o conjunto proposto foi capaz de superar o *regular normal patterns* em 70,8% e o *meet-in-the-middle* em 43,4%, na média. Isso abre novas possibilidades de pesquisas, não apenas em termos de formulações para o problema do empacotamento em bins, mas também de generalizações para problemas de dimensões mais altas.

PALAVRAS CHAVE. Problema de Empacotamento em Bins. Conjunto de Pontos. Meet-in-the-middle Patterns.

Tópicos: OC - Otimização Combinatória; PM - Programação Matemática.

ABSTRACT

The bin packing problem has a large number of applications. Some of its solution methods are based on the positioning of items, as arc-flow formulations. Approaches for reducing

https://proceedings.science/p/106754?lang=pt-br

the number of positions at which items can be arranged have been proposed since the seventies until very recently with the sets of meet-in-the-middle patterns. In this paper, we propose a new approach, based on an arbitrary ordering of items, that results in an improved set of points. The idea consists of eliminating redundant positions by right- and left-aligning items in solutions. The computational experiments carried out on benchmark instances have shown that the proposed sets outperformed the regular normal patterns in 70.8% and the meet-in-the-middle in 43.4%, on average. This opens new possibilities of future research, not only in terms of formulations for the bin packing problem but also in terms of generalization for problems of higher dimensions.

KEYWORDS. Bin Packing Problem. Set of Points. Meet-in-the-middle Patterns.

Topics: CO - Combinatorial Optimization; MP - Mathematical Programming.

1. Introduction

The *Bin Packing Problem* (BPP) has several applications, as in logistics, manufacturing, and computing. Some applications arise in cutting and packing of objects, scheduling of jobs, routing of vehicles with loading constraints, among others [Wäscher et al., 2007; Pereira, 2016; Wei et al., 2018]. Recently, [Delorme et al., 2016] surveyed solution methods for the BPP and the *Cutting Stock Problem* (CSP), claiming the need of exact methods able to solve large-sized instances in a satisfactory computational time, once these problems are NP-hard [Garey and Johnson, 1979].

In the Bin Packing Problem (BPP), there is an unlimited number of copies of a one-dimensional bin with a given capacity, and a finite set of one-dimensional items, each one with a given size. The aim of this problem is to pack all items into the minimum number of bins, such that items packed in a bin has total size at most the capacity of the bin. Among the solution methods for the BPP, in the sixties, [Gilmore and Gomory, 1961, 1963] proposed a column generation approach which in turn is used to solve the linear relaxation of a set-covering formulation. In the well-known book "Knapsack Problems" by [Martello and Toth, 1990], a chapter is dedicated to the BPP presenting solution methods and computational experiments.

[Dyckhoff, 1981] presented an Integer Linear Programming model for the BPP, where variables are related to positions (i.e., points) where to cut the bin in order to produce items. Another formulation based on the positioning of items is the arc-flow model [Valério de Carvalho, 1999]. This formulation considers vertices, which are equivalent to points in a bin, and arcs, which represent the portion of a bin that items occupy. Thus, a path of vertices and arcs represent a packing in a bin. This author proposed some reduction procedures in order to eliminate redundant vertices and arcs, as well as proved the lower bound of the linear relaxation of the arc-flow formulation is equivalent to the one obtained via the set-covering formulation in [Gilmore and Gomory, 1961, 1963]. Recently, [Brandão and Pedroso, 2016] generalized the arc-flow formulation and showed how to solve different cutting and packing problems with this formulation.

Despite of the equivalence between the lower bounds obtained from the arc-flow and the set-covering formulations, the arc-flow formulation has a pseudo-polynomial number of variables and constraints, while the set-covering formulation has an exponential number of variables. For practical (i.e., large-sized) instances, in the arc-flow formulation, the number of vertices and arcs can be very large and thus this formulation cannot be exactly solved within a reasonable computational time. Moreover, one conjecture that is related with the set-covering (and, consequently the arc-flow) formulation is the Modified Integer Round Up Property (MIRUP): for any instance of the BPP, the optimal solution of this instance is at most $\lceil z_{lb} \rceil + 1$, where z_{lb} is the value of an optimal solution for the relaxation of the set-covering formulation. In practice, the bound given by the relaxation of this formulation is even tighter since for many instances the value of its optimal solution is equal to $\lceil z_{lb} \rceil$. In fact, it is not straightforward to find instances where the value of the optimal solution of the linear relaxation is equal to $\lceil z_{lb} \rceil + 1$ [Caprara et al., 2015; Kartak et al., 2015].

Approaches for obtaining smaller sets of points for the positioning of items have been proposed since the seventies. To the best of our knowledge, [Herz, 1972] and [Christofides and Whitlock, 1977] proposed independently the first sets of points, in the context of a two-dimensional orthogonal cutting problem. These sets were called, respectively, canonical dissections and normal patterns, however, they are the same. [Terno et al., 1987] introduced the sets of reduced raster points, which generally is smaller than the sets of normal patterns, while [Boschetti et al., 2002] presented a way of reducing the sets of normal patterns by computing sets of regular normal patterns, which are related to each item individually. Recently, [Côté and Iori, 2018] proposed the sets of meet-in-the-middle patterns that can outperform all previous sets already published in the literature.

https://proceedings.science/p/106754?lang=pt-br

In the present paper, we propose new sets based on an arbitrary ordering of items that was able to, on average, reduce by more than 60% the meet-in-the-middle patterns. The remaining of the paper is organized as follows. In Section 2, we present a theoretical background on the BPP and sets of points. Next, Section 3 presents our approach, while computational experiments are discussed in Section 4. Finally, in Section 5, the concluding remarks and ideas for future works are presented.

2. Theoretical Background

In the BPP, there are unlimited copies of a one-dimensional bin whose size (or capacity) is $W \in \mathbb{Z}_+$, and a set I of one-dimensional items, such that each item $i \in I$ has size (or weight) $w_i \in \mathbb{Z}_+$. The objective is to pack all items of I into the minimum number of bins, such that the capacity of the bins are not exceeded. A bin is seen as an horizontal line segment which can be discretized in the set of points $\{0, 1, \ldots, W\}$. An item i is packed in the point p if it starts at p and ends at $p + w_i$. Notice that i can be packed in the points $\{0, \ldots, W - w_i\}$.

In the following, some sets of points for the positioning of items into the bin are presented. Although some of those sets were proposed in the context of two-dimensional problems, all of them can be applied without loss of optimality to solve the BPP.

The set of *Canonical Dissections* was proposed by [Herz, 1972] and preserves an optimal solution of the problem. It is an improvement over the naive unitary discretization of the bin. The same set of points was independently proposed by [Christofides and Whitlock, 1977] with the name of *Normal Patterns*. This set of points is based on the idea that any item of a packing can be left-aligned until it touches either the beginning of the bin or the right-side of an item. Then, the set N of normal patterns is obtained from conic combinations (i.e., non-negative binary linear combinations of the items' weight), that is:

$$N = \{p \mid p = \sum_{j \in I} \zeta_j w_j \leq W, \zeta_j \in \{0, 1\}, \forall j \in I\}. \tag{1}$$

[Terno et al., 1987] proposed an improvement over the set of normal patterns, the so-called *Reduced Raster Points*. This set improves the normal patterns by calculating the maximum portion of the bin that can be occupied after each normal pattern, so eliminating redundant points. Then, the set R of raster points is obtained by:

$$R = \{\langle W - p \rangle \mid \langle W - p \rangle = \max\{c \in N \mid c \leq W - p\}\}. \tag{2}$$

To the best of our knowledge, [Boschetti et al., 2002] was the first to propose an approach that generates sets of points for each item individually. The resulting sets are called *Regular Normal Patterns*. These authors observed that an item $i \in I$ does not need to be part of the conic combinations that generate it. Hence, the set B_i of regular normal patterns is obtained from the conic combinations in which i is not considered, that is:

$$B_i = \{p \mid p = \sum_{j \in I \setminus \{i\}} \zeta_j w_j \leq W - w_i, \zeta_j \in \{0, 1\}, \forall j \in I \setminus \{i\}\}. \tag{3}$$

Recently, [Côté and Iori, 2018] proposed the sets of *Meet-in-the-Middle Patterns* that are generated from a given parameter $t \in \{0, 1, \ldots, W\}$ and for each item $i \in I$. The idea is to left align all items that are packed in points at the left of t, as well as to right align all items that are positioned at the right of t. Thus, two sets are calculated for an item i, that is, L_i has the points that are at the left of t, for left-aligned packings, and R_i has the points that are at the right of t,

https://proceedings.science/p/106754?lang=pt-br

for right-aligned packings. The resulting set M_i of meet-in-the-middle patterns, for an item i, is described in (6).

$$L_i = \{p \mid p \leq \min\{t, W - w_i\}, p \in B_i\}, \tag{4}$$

$$R_i = \{W - w_i - p \mid t < W - w_i - p, p \in B_i\}, \tag{5}$$

$$M_i = L_i \cup R_i. \tag{6}$$

[Côté and Iori, 2018] proposed a pseudo-polynomial algorithm that has worst case time-complexity $O(|I|^2 W)$ to determine the value of t that minimizes $|\cup_i M_i|$ (e.g., in the arc-flow model, this is advantageous for reducing the number of constraints) or $\sum_i |M_i|$ (e.g., in the arc-flow model, this is advantageous for reducing the number of variables). These authors also proposed two preprocessing techniques to reduce the size of the sets while preserving an optimal solution of the problem. The first preprocessing is based on fixing at which side of the bin the smallest item in I is going to be packed. The second preprocessing is based on eliminating redundant points by checking, for pairs of points $p, q \in M_i$, where $p < q$, for every feasible packing in a bin with the item i at p if there is an equivalent feasible packing with i at q, then p can be removed from M_i.

Other than the two preprocessing, [Côté and Iori, 2018] also proposed further reductions of the meet-in-the-middle when solving the BPP, which are similar to those procedures in [Valério de Carvalho, 1999]. Notice that in the BPP, the order in which the items are packed in a bin is irrelevant. [Côté and Iori, 2018] considered that items can be packed respecting a non-increasing order of their size. In this way, the sets B_i, when calculating M_i, can be redefined as B'_i, that is:

$$B'_i = \{p \mid p = \sum_{j<i} \zeta_j w_j \leq W, \zeta_j \in \{0,1\}, \forall j < i\}. \tag{7}$$

3. Proposed Set of Points

In the BPP, the order in which the items are packed in each bin does not impact on the optimal solution. Observing this principle, we can derive further reductions on the sets of points based on fixing a given order for items. Therefore, considering the BPP, in which items are organized in an arbitrary order (e.g., sorted into non-increasing order of items' size) and thus packed respecting such order (which we call *ordering criterion*), we propose an approach to obtain new sets of points for each item $i \in I$, without loss of optimality.

For each pair of items $i, j \in I, i < j$ means that in every feasible packing, if i and j are in the same bin, then i is packed to the left of j. Moreover, for an item i and point $p \in \{0, 1 \ldots, W\}$, the value K^L_{ip} in (8) represent the maximum size of a feasible packing with items to the left of i, that is at most p. Similarly, the value K^R_{ip} is defined in (9) for the feasible packing with items to the right of i.

$$K^L_{ip} = \max\{s \mid s = \sum_{j<i} \zeta_j w_j \leq p, \zeta_j \in \{0,1\}, \forall j < i\}. \tag{8}$$

$$K^R_{ip} = \max\{s \mid s = \sum_{j>i} \zeta_j w_j \leq p, \zeta_j \in \{0,1\}, \forall j > i\}. \tag{9}$$

Let S_i be a set of points related with item $i \in I$ (e.g., normal patterns). Considering the ordering of items, the improved set $O^R_i(S_i)$ in (10) is obtained by moving the points of S_i as much

to the right as possible, such that, contrary to the normal patterns, each packing is now right-aligned. On the other hand, in the improved set $O_i^L(S_i)$ in (11), the points of S_i are moved as much to the left as possible, as in the normal patterns, obtaining a packing that is left-aligned to the bin.

$$O_i^R(S_i) = \{W - w_i - K^R_{i(W-w_i-p)} \mid \forall p \in S_i\}. \tag{10}$$

$$O_i^L(S_i) = \{K^L_{ip} \mid \forall p \in S_i\}. \tag{11}$$

Algorithm 1 computes the improved sets $O_i^R(S_i)$, while Algorithm 2 obtains the improved sets $O_i^L(S_i)$.

Algorithm 1: Algorithm to compute the improved sets $O_i^R(S_i)$.

Input: the bin size W, and the size w_i and a set of points S_i, for each item $i \in I$.
Output: the improved sets of points $O_i^R(S_i)$, for each item $i \in I$.

1 $K^R \leftarrow$ matrix of order $|I| \times W$;
2 **for** $i \leftarrow 1, \ldots, |I|$ **do**
3 $\quad O_i^R \leftarrow \emptyset$;
4 **end**
5 **for** $p \leftarrow 0, \ldots, W$ **do**
6 $\quad K^R_{|I|p} \leftarrow 0$;
7 **end**
8 **for** $i \leftarrow |I|, \ldots, 1$ **do**
9 $\quad$ **for** $p \leftarrow 0, \ldots, W$ **do**
10 $\quad\quad$ **if** $p < w_i$ **then**
11 $\quad\quad\quad K^R_{ip} \leftarrow K^R_{(i+1)p}$;
12 $\quad\quad$ **else**
13 $\quad\quad\quad K^R_{ip} \leftarrow \max\{K^R_{(i+1)p}, K^R_{i(p-w_i)} + w_i\}$;
14 $\quad\quad$ **end**
15 $\quad$ **end**
16 **end**
17 **for** $i \leftarrow 1, \ldots, |I|$ **do**
18 $\quad$ **for** $p \in S_i$ **do**
19 $\quad\quad O_i^R \leftarrow O_i^R \cup \{W - w_i - K_{i(W-w_i-p)}\}$;
20 $\quad$ **end**
21 **end**
22 **return** $O^R = \{O_1^R, \ldots, O^R_{|I|}\}$;

In Algorithm 1, the variable K^R is a matrix whose number of rows is the number of items and the number of columns is the size of the bin. As no item can be packed at the right of the last item $|I|$ due to the ordering criterion, then, $K^R_{|I|p} = 0$ (lines $5-7$). The computation of K^R (lines $8-16$) is based on the dynamic programming algorithm for the one-dimensional knapsack problem [Martello and Toth, 1990]. Moreover, notice that all items which can be packed at the right of item $i+1$, they can also be packed at the right of i. Therefore, the computation of K_i^R (lines $10-14$) considers the value in $K^R_{(i+1)}$. At the lines $17-21$, the improved set O_i^R holds, for each point $p \in S_i$, the point $W - w_i - K_{ip}$ as stated in (10). Similarly, Algorithm 2 compute the

improved sets $O_i^L(S_i)$. There are two main differences with the algorithm for $O_i^R(S_i)$, which are: the order in which the items are visited when computing K^L (the loop that begins at the line 8), and the computation of the improved set O_i^L (line 19), which follows (11).

Algorithm 2: Algorithm to compute the improved sets $O_i^L(S_i)$.

Input: the bin size W, and the size w_i and a set of points S_i, for each item $i \in I$.
Output: the improved sets of points $O_i^L(S_i)$, for each item $i \in I$.

```
1  K^L ← matrix of order |I| × W;
2  for i ← 1, ..., |I| do
3  |   O_i^L ← ∅ ;
4  end
5  for p ← 0, ..., W do
6  |   K^L_{|I|p} ← 0;
7  end
8  for i ← 1, ..., |I| do
9  |   for p ← 0, ..., W do
10 |   |   if p < w_i then
11 |   |   |   K^L_{ip} ← K^L_{(i−1)p};
12 |   |   else
13 |   |   |   K^L_{ip} ← max{K^L_{(i−1)p}, K^L_{i(p−w_i)} + w_i};
14 |   |   end
15 |   end
16 end
17 for i ← 1, ..., |I| do
18 |   for p ∈ S_i do
19 |   |   O_i^L ← O_i^L ∪ {K_{ip}};
20 |   end
21 end
22 return O^L = {O_1^L, ..., O_{|I|}^L};
```

Both Algorithms 1 and 2 have the same worst-case time complexity $O(|I|W)$. For an item $i \in I$, we propose to first apply O_i^R to the set S_i, that is, to align all points to the right, and then to apply O_i^L to the resulting set, that is, re-aligning the resulting points to the left. Therefore, the final set is given as $O_i = O_i^L(O_i^R(S_i))$.

In order to illustrate how to obtain the improved sets, we consider a numerical example in which the bin has size $W = 10$ and four items in I, such that $w_1 = 6$, $w_2 = 5$, $w_3 = 3$, $w_4 = 2$. We consider S_i as the set B_i' in (7), such that: $B_1' = \{0\}$, $B_2' = \{0\}$, $B_3' = \{0, 5, 6\}$, and $B_4' = \{0, 3, 5, 6, 8\}$. Notice that $|\cup_i B_i'| = 5$ and $\sum_i |B_i'| = 10$. Applying O^R to the sets B', we first compute K^R, where K_{ip}^R represents the maximum sum that is smaller than or equal to p of the items' size that can be packed at the right of i (every item $j > i$). Then, the matrix K^R is, for the rows indexed as $1, \ldots, 4$ and the columns as $0, \ldots, 10$:

$$K^R = \begin{bmatrix} 0 & 0 & 2 & 3 & 3 & 5 & 5 & 7 & 8 & 8 & 10 \\ 0 & 0 & 2 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For item 1, when it is packed on 0, the maximum sum of items' size that can be at its right is $K^R_{1,W-w_1-0} = K^R_{1,4} = 3$. As at the end of the algorithm every packing is right-aligned, the point 0 can be pushed to $W - w_i - K^R_{1,W-w_1-0} = 10 - 6 - K^R_{1,4} = 1$ and then $O^R_1(B'_1) = \{1\}$. Similarly, for item 2, the point 0 can be pushed to $W - w_2 - K^R_{2,W-w_2-0} = 10 - 5 - 5 = 0$ and then $O^R_2(B'_2) = \{0\}$. For item 3, the points 0, 5, and 6 can be pushed, respectively, to $W - w_3 - K^R_{3,W-w_3-0} = 5$, $W - w_3 - K^R_{3,W-w_3-5} = 5$, and $W - w_3 - K^R_{3,W-w_3-6} = 7$, and the resulting set is $O^R_3(B'_3) = \{5, 7\}$. Finally, for item 4, as no item is going to be packed at its right, then $K^R_{4,p} = 0$, for every $p = \{0, \ldots, W\}$. Therefore, $W - w_4 - K^R_{4,W-w_4-p} = 10 - 2 - 0 = 8$ for every $p \in B'_4$, and each point in B'_4 is pushed to 8 and then $O^R_4(B'_4) = \{8\}$. The resulting sets are $O^R_1(B'_1) = \{1\}, O^R_2(B'_2) = \{0\}, O^R_3(B'_3) = \{5, 7\}$, and $O^R_4(B'_4) = \{8\}$, and then $|\cup_i O^R_i(B'_i)| = 5$ and $\sum_i O^R_i(B'_i) = 5$. Now, it is applied O^L to the resulting sets in order to left align the points. In this case, the K^L is calculated similarly to K^R, where K^L_{ip} denotes the maximum sum that is smaller than or equal to p of the items' size that can be packed at the left of i (every item $j < i$). The matrix K^L is:

$$K^L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 6 & 6 & 6 \\ 0 & 0 & 0 & 0 & 0 & 5 & 6 & 6 & 6 & 6 & 6 \\ 0 & 0 & 0 & 3 & 3 & 5 & 6 & 6 & 8 & 9 & 9 \end{bmatrix}$$

In the computation of the sets O^L, we pushed the points as much to the left as possible. For item 1, as no item can be pushed to the left of it, all its points can be moved to 0. Notice that the only point in $O^R_1(B'_1)$ is 1 and then $K^L_{1,1} = 0$. For item 2, the only point is 0 and then $K^L_{2,0} = 0$. For item 3, the current points are 5 and 7, and as $K^L_{3,5} = 5$, $K^L_{3,7} = 6$, then the point remains, whereas 7 is pushed to 6. Finally, for item 4, there is only the point 8, and as $K^L_{4,8} = 8$, then such point remains. It follows that the final improved sets are $O_1 = O^L_1(O^R_1(B'_1)) = \{0\}$, $O_2 = O^L_2(O^R_2(B'_2)) = \{0\}$, $O_3 = O^L_3(O^R_3(B'_3)) = \{5, 6\}$, $O_4 = O^L_4(O^R_4(B'_4)) = \{8\}$, and for such sets, $|\cup_i O_i| = 4$ and $\sum_i |O_i| = 5$. Figure 1 has the arc-flow representation of the sets related to this example.

4. Computational Experiments

Computational experiments were carried out in order to compare the proposed sets of points with the state-of-the-art sets (i.e., the sets of meet-in-the-middle patterns with the preprocessing and the non-increasing order for the items) when generating the sets of points for the BPP. All the algorithms were implemented in the C++ programming language and were applied on benchmark instances that are available at the Bin Packing Problem Library proposed by [Delorme et al., 2018]. The experiments were performed on a computer with a Xeon X3430 processor at 2.40 GHz, 12 GB of RAM, and the Ubuntu 14.04.5 as operational system. It is worth mentioning that the computational time is not reported because it is negligible for all instances, however, notice that the proposed sets are calculated in $O(|I|W)$ while the state-of-the-art sets are calculated in $O(|I|^2W)$.

Table 1 shows information of nine classes, comprising 1607 instances. Each line of this table holds the name of the class, the total number of instances in the class (i.e., $\#Instances$), the

https://proceedings.science/p/106754?lang=pt-br

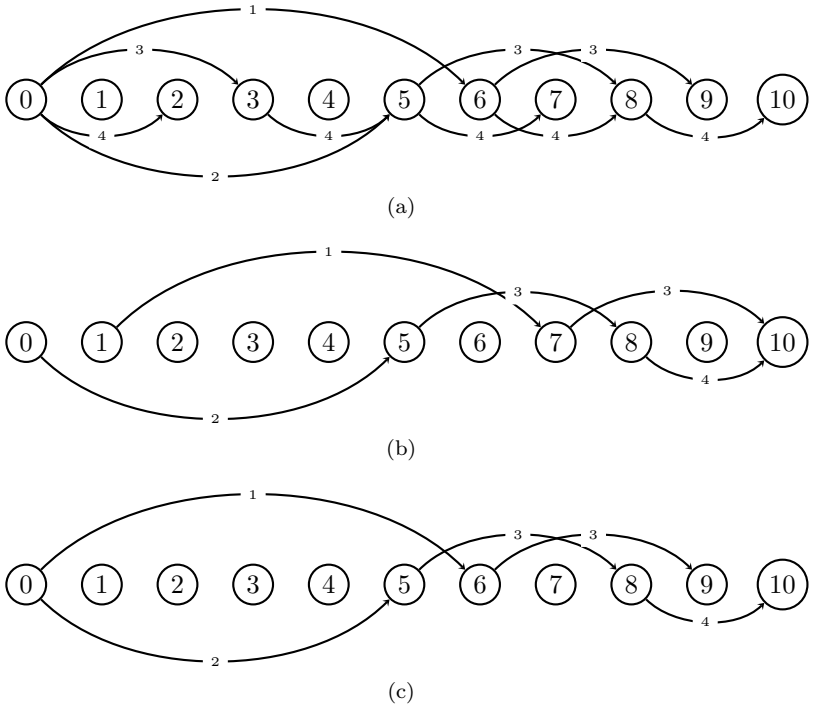


Figure 1: (a) Representation of B'; (b) Representation of $O^R(B')$; (c) Representation of $O^L(O^R(B'))$.

average number of items in the class (i.e., $\overline{|I|}$), the average size of the bin in the class (i.e., $\overline{W}$), and the results regarding the sets of regular normal patterns B'_i for the BPP, in (7), in the class. Then, it is presented the average size of the set B' (i.e., $\overline{|\cup_i B'_i|}$, which represents the average number of constraints the arc-flow formulation for the BPP has) and the average size of the overall set B' when considering all individual sets for the items (i.e., $\overline{\sum_i |B'_i|}$, which represents the average number of variables the arc-flow formulation for the BPP has).

Table 1: Information of instances and results for the sets of regular normal patterns for the BPP.

Class	# Instances	$\overline{\lvert I\rvert}$	$\overline{\lvert W\rvert}$	$\overline{\lvert \cup_i B'_i\rvert}$	$\overline{\sum_i \lvert B'_i\rvert}$
AI201	50	171	2,220	2,068.8	177,050.3
ANI201	50	170	2,220	2,064.8	174,984.6
Falkenauer_T	80	118	1,000	367.3	44,834.6
Scholl 1	720	64	123	85.2	5,513.1
Scholl 2	480	98	1,000	744.8	86,811.0
Scholl 3	10	199	100,000	28,785.0	1,486,281.3
Schwerin 1	100	44	1,000	535.4	24,993.6
Schwerin 2	100	46	1,000	540.0	30,102.1
Wascher	17	50	10,000	9,063.9	601,236.7

In Table 2, it is shown the reduction that is achieved with the state-of-the-art and the pro-

Table 2: Reductions (in percentage) from the set B'.

	Reductions from $\lvert \cup_i B'_i \rvert$		Reductions from $\sum_i \lvert B'_i \rvert$	
Class	MIM-BPP	Proposed Sets	MIM-BPP	Proposed Sets
AI201	6.8%	**15.4%**	24.5%	**55.9%**
ANI201	6.9%	**15.4%**	24.6%	**55.7%**
Falkenauer_T	39.2%	**69.5%**	80.8%	**94.0%**
Scholl 1	7.1%	**21.5%**	53.9%	**57.4%**
Scholl 2	12.1%	**25.7%**	43.9%	**52.0%**
Scholl 3	80.1%	**93.5%**	98.1%	**99.3%**
Schwerin 1	29.0%	**65.5%**	61.2%	**84.2%**
Schwerin 2	28.4%	**64.9%**	61.0%	**83.8%**
Wascher	10.2%	**32.6%**	42.6%	**54.5%**
Average:	24.4%	**44.9%**	54.5%	**70.8%**

posed sets, for each class of instances, when comparing with the sets of regular normal patterns for the BPP in Table 1. Concerning the state-of-the-art sets, we follow [Côté and Iori, 2018] and consider the MIM-BPP, which represents the meet-in-the-middle patterns with the second preprocessing and the non-increasing order for the items. In both cases, the sets are defined in terms of minimizing $\sum_i |M_i|$. Regarding the computation of the reductions, for a given sets of points S_i, two metrics are considered, which are: $\frac{|\cup_i B'_i| - |\cup_i S_i|}{|\cup_i B'_i|}$, which represents the reduction in the number of constraints of the arc-flow formulation for the BPP; and, $\frac{\sum_i |B'_i| - \sum_i |S_i|}{\sum_i |B'_i|}$, which represents the reduction in the number of variables.

Observing the results of Table 2, concerning the reductions in the number of constraints (i.e., the first metric), the MIM-BPP holds an overall average reduction by 24.4% in comparison with the regular normal patterns for the BPP. However, this set is not able to outperform the proposed sets whose reductions are better for all classes, with an overall average reduction by 44.9%. Regarding the reductions in the number of variables (i.e., the second metric), the MIM-BPP has an overall average reduction by 54.5% in comparison with the regular normal patterns for the BPP. Once again, the proposed sets could outperform the MIM-BPP, with an overall average reduction by 70.8% in comparison with the regular normal patterns for the BPP. Moreover, for four out of nine classes, the proposed sets reduced the regular normal patterns for the BPP above the 80.0%, while for the other classes, the reduction was always greater than 50.0%.

Further analysis is drawn in order to show how the proposed sets are effective in comparison with the state-of-the-art MIM-BPP. Table 3 has the percentage reductions by the proposed sets from the MIM-BPP, for each class, considering the number of constraints (i.e, $\lvert \cup_i M_i \rvert$), $\frac{|\cup_i M_i| - |\cup_i O_i|}{|\cup_i M_i|}$, and the number of variables (i.e., $\sum_i |M_i|$), $\frac{\sum_i |M_i| - \sum_i |O_i|}{\sum_i |M_i|}$. Concerning the number of constraints, the reductions range from 9.2% (class ANI201) to 67.3% (class Scholl 3), with an overall reduction by 33.2%. In terms of the number of variables, the overall reduction is by 43.4%, while the smaller reduction is observed on the class Scholl 1 and it corresponds to 12.7%. The best reduction belongs to the class Falkenauer_T and it is by 67.4%.

https://proceedings.science/p/106754?lang=pt-br

Table 3: Reductions (in percentage) from the MIM-BPP.

Class	Reductions from $\lvert \cup_i M_i \rvert$	Reductions from $\sum_i \lvert M_i \rvert$
AI201	9.4%	41.6%
ANI201	9.2%	41.3%
Falkenauer_T	49.9%	67.4%
Scholl 1	16.2%	12.7%
Scholl 2	17.7%	19.4%
Scholl 3	67.3%	65.4%
Schwerin 1	51.4%	59.2%
Schwerin 2	51.0%	58.4%
Wascher	26.7%	25.1%
Average:	33.2%	43.4%

5. Concluding Remarks

It is proposed a new set of points that is associated with formulations based on the positioning of items for the bin packing problem. The proposed set is able to outperform the state-of-the-art meet-in-the-middle patterns even in the presence of preprocessing techniques and further reductions for the bin packing problem.

While the state-of-the-art sets requires the worst-case time complexity $O(|I|^2W)$ to be obtained, the proposed sets have the same worst-case time complexity that is required by the normal patterns, that is, $O(|I|W)$. Moreover, the computational experiments on benchmark instances have indicated that the proposed sets can reduce the size of formulations based not only on the regular normal patterns for the BPP (by 44.9%, in the number of constraints, and 70.8%, in the number of variables), but also in comparison with the state-of-the-art sets (by 33.2%, in the number of constraints, and 43.4%, in the number of variables).

As future research, we will consider the evaluation of the proposed sets when solving the arc-flow formulation for the bin packing problem. Another direction is related to extend the proposed sets to solve the contiguous bin packing problem, and cutting and packing problems with two and three dimensions.

References

Boschetti, M., Hadjiconstantinou, E., and Mingozzi, A. (2002). New upper bounds for the two-dimensional othogonal non guillotine cutting stock problem. *IMA Journal of Management Mathematics*, 13:95–119.

Brandão, F. and Pedroso, J. (2016). Bin packing and related problems: General arc-flow formulation with graph compression. *Computers & Operations Research*, 69:56 – 67.

Caprara, A., Dell'amico, M., Díaz-Díaz, J., Iori, M., and Rizzi, R. (2015). Friendly bin packing instances without integer round-up property. *Mathematical Programming*, 150(1):5–17.

Christofides, N. and Whitlock, C. (1977). An algorithm for two-dimensional cutting problems. *Operations Research*, 25(1):30–44.

Côté, J.-F. and Iori, M. (2018). The meet-in-the-middle principle for cutting and packing problems. *INFORMS Journal on Computing*, 30(4):646–661.

https://proceedings.science/p/106754?lang=pt-br

Delorme, M., Iori, M., and Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1):1–20.

Delorme, M., Iori, M., and Martello, S. (2018). BPPLIB: a library for bin packing and cutting stock problems. *Optimization Letters*, 12(2):235–250.

Dyckhoff, H. (1981). A new linear programming approach to the cutting stock problem. *Operations Research*, 29(6):1092–1104.

Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: Freeman.

Gilmore, P. and Gomory, R. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9:849–859.

Gilmore, P. and Gomory, R. (1963). A linear programming approach to the cutting stock problem - part II. *Operations Research*, 11:863–888.

Herz, J. (1972). Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development*, 16(5):462–469.

Kartak, V., Ripatti, A., Scheithauer, G., and Kurz, S. (2015). Minimal proper non-IRUP instances of the one-dimensional cutting stock problem. *Discrete Applied Mathematics*, 187(C):120–129.

Martello, S. and Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA. ISBN 0-471-92420-2.

Pereira, J. (2016). Procedures for the bin packing problem with precedence constraints. *European Journal of Operational Research*, 250(3):794–806.

Terno, J., Lindemann, R., and Scheithauer, G. (1987). Zuschnittprobleme and ihre praktische lösung. Technical report, Verlag Harry Deutsch, Thun und FrankfurtMain.

Valério de Carvalho, J. (1999). Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research*, 86(0):629–659.

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.

Wei, L., Zhang, Z., Zhang, D., and Leung, S. C. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 265(3):843–859.