Improving the 1-Bounded Space Algorithms for 2-Dimensional Online Bin Packing

Xarnach Rotllan, Ferran

Todays Date

Abstract

In this paper we study the 1-bounded space of 2-dimensional bin packing. A sequence of rectangular items arrive one at a time, and the following item arrives only after the packing of the previous one, which after being packed cannot be moved. The bin size is 1×1 and the width and height of the items are ≤ 1. The objective is to minimize the number of bins used to pack all the items. At any time there is only 1 active bin, and the previously closed bins cannot be used for any subsequent items. The new algorithm offers an improvement of the previous best known 8.84-competitive algorithm to a 6.53-competitive, it also raises the lower bound from 2.5 to $2.\hat{6}$.

1 Introduction

In the bin packing problem, a sequence of items are packed into different bins of the same size without overlapping. The objective is to minimize the number of bins used to pack *all* the items. The size of the items are at most the size of a bin.

In the *online* variation of the bin packing problem, the items have to be packed as they arrive, and they cannot be moved afterwards. In the online *bounded space* variation, the items can only be packed into the active bins. When none of the active bins can pack the newly arrived item, one active bin has to be closed and a new bin has to be opened and labeled *active*.

The online bounded space variation on a 1-dimensional space, has been subject of study for long time. Csirik. et al. [2] showed that simple K-Bounded Best Fit BBF_k (base on the Best-fit algorithm using bounded space) is better than Next-K fit algorithms (Based on the First-fit algorithm), with an asymptotic worst-case ratio of 17/10 for all $k \geq 2$ and better packings on average. Lee. et al. [3] created an harmonic algorithm that performs better on the worst case (but poorly on the average) with a worst-case ratio of 1.69103.

An interested variation is the one with only one active bin at a time (1-bounded space) on a 2-dimensional space. This variant originates from grid computing, in which there is a cluster of computers arranged on a grid (the bin) and each job requests a rectangular subgrid of computers (the rectangular items) which can only be rotated by 90°, and as soon as a subgrid has been assigned it cannot be moved. There is only one grid machine and the jobs arrive online. The objective is to pack as many jobs as possible into this fixed size grid. When all the jobs are finished, another set of jobs can be packed.

This paper is regarding the online variation of 1-bounded space 2-dimensional bin packing allowing the 90° rotation of items [1]. The items arrive online, and are packed without knowledge of the shape of subsequent items. Once an item has been packed, it cannot be moved.

The size of the square bin is 1×1, and the sides of every item are smaller than 1. Since 90° rotation is allowed, and x being the width of an item and y its height, w.l.o.g. assume that $x \geq y$. The space occupied by an item is the product xy, and the total occupied space in a bin is the sum of the total occupied space of the items in it. The problem is to maximize the occupied space per bin, packing all the items without overlaps.

To evaluate an online algorithm used for bin packing, the *asymptotic competitive ratio* which is defined as follows is used. Consider an online algorithm A and an optimal offline algorithm OPT. For a sequence S of items, let $A(S)$ be the cost (number of bins used) of the algorithm A and let $OPT(S)$ be the optimal cost of OPT. Then the *asymptotic competitive ratio* for A is:

$$R_A^\infty = \lim_{k\to\infty} \sup_S \{\frac{A(S)}{OPT(S)} | OPT(S) = k\}.$$

The previous results on this variant by Fujita [6] consist on a lower bound of 23/11 and an upper bound $O((\log\log m)^2)-$comptetitive, where m is the width of the square bin, and it is of value ≥ 512. Those are improved by Chin. et al. [1], raising the lower bound to 2.5 and making the upper bound independent from the size of the bin, getting an 8-competitive algorithm.

In the algorithm by Chin. et al. [1], the items are classified into three classes $\{A : x \geq 1/2\}$, $\{B : 1/6 \leq x < 1/2\}$ and $\{C : x < 1/6.\}$ (assuming $x \geq y$), and the square bin is partitioned into two areas U and L. Big items (A and B classes) are packed into area U of the bin, and the small items into area L, following two different strategies for U and L.

The worst case scenario of this algorithm is when one area of the bin (U or L) is empty and the bin is closed because a newly arrived item cannot be packed into the other area. In this scenario the inefficiency of a strict grouping into big and small items (U and L areas) becomes obvious. The improved algorithm aims at lowering the importance of the distinction of the items characteristics, thus raising the amount of used space per bin.

Our contribution is the raise of the lower bound up to $2.\hat{6}$ and an improved algorithm offers a worst-case ratio of 6.53. On the improved algorithm, the items are also classified in the three clases the algorithm by Chin. et al. is

[1] It's easy to see that if rotation is not allowed, no constant bounds could exist as proved by Fujita [6], because two perpendicular rectangles of sizes $1 \times \epsilon$ could never be packed into the same bin.

classified, and the bin initially is divided into the same two areas, when a small (C class) item cannot be packed, if there is space enough in U a new area is allocated there to pack small items as explained in section 2.3. The areas of the bin are named differently in the new algorithm, because is not important to distinguish which area is the upper as more than one area for small items can exist. The area which big items are packed into, is named **B** and the area into which small items are packed into are named **S**. This naming will be used to explain the previous best constant-competition algorithm as well.

2 An improved Algorithm

In order to achieve a better constant-competitive algorithm, we are going to apply a modification into the previous best constant-competitive algorithm [1]. In that algorithm the rectangular items are classified into three classes A, B and C with $x \geq y$ so that

$$\begin{aligned} A &= \{(x,y)|x \geq 1/2\}, \\ B &= \{(x,y)|1/6 \leq x < 1/2\}, \text{and} \\ C &= \{(x,y)|x < 1/6\}. \end{aligned}$$

From now on, items will be called by their class: A-items, B-items and C-items.

Each square bin is divided into a bottom **B**-area (for A- and B-items) and a top **S**-area. Items are packed in **S** and **B** following two different strategies as explained in sections 2.1 and 2.2. When a newly arrived item a cannot be packed into its designated area, the active bin is closed and another one is opened to pack item a.

In the improved algorithm the packing strategies for **B** and **S** are the same and two initial **B**- and **S**-areas are defined, but depending on the items that keep comming, different **S**-areas can be allocated inside of the original **B**-area. How those **S**-areas are allocated is explained in section 2.3.

This allows a bigger initial **B**-area on the improved algorithm, resulting in a better asymptotic competitive ratio.

2.1 Packing Strategy for Area B

A-items are packed following a top-down order, and the vertical symmetry axis of each item aligns with the vertical symmetry axis of the square bin. B-items are packed by a bottom-up order in both the left and the right side of **B**, while balancing the height of both sides, e.g., the new B-item a is packed into the side with the lower height. For instance in figure 1 a newly arrived item a should be packed into the right area, considering $y_1 \geq y_2$.

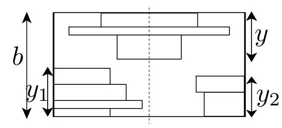

Figure 1: Packing big items into **B**.

2.2 Packing Strategy for Area S

Define a C item as area of the subclass C_i if $2^{-i-1}/6 < x \leq 2^{-i}/6$, and w_i as the biggest possible width of all items from subclass C_i, i.e., $w_0 = 1/6$, $w_1 = 1/12$. According to this, the area for small items can be partitioned into columns of width $w_i (i > 0)$ [2]. For every subclass C_i, define col_i to be the active column, i.e., the only column of size w_i currently open.

Let f be the width of the free space at the right part of **S**. A newly arrived item $a \in C_i$ is packed into col_i following a bottom-up order. If a cannot fit into col_i, this column is closed and another column is created and labeled active in the free space right of the last column, in order to pack a. If a new column cannot be created, the active **S** is closed.

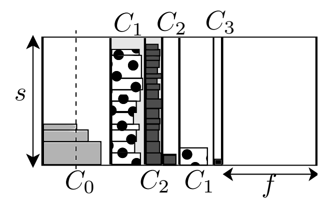

Figure 2: Packing small items into **S**.

E.g., in figure 2, we can see that there are four subclasses C_i of items. The second and third columns had been closed because the items in columns four and five were to big to fit in there, thus, these columns will not be used anymore [3]. The active columns are the first, fourth, fifth and sixth. If a newly arrived item a is of one of the subclases C_i ($i < 4$) then it can be packed into one of the existent columns, otherwise a new column would have to be created in the f amount of free space on the right of the last column. Note in col_0 that all the items have a width larger than $w_o/2$, this is true for all the columns by the definition of C_i.

2.3 Allocation of S-areas in the Improved Algorithm

The improved version of this algorithm on the other hand defines a much smaller initial **S**-area in a bin, but the final space used for small items can grow during

[2]The number of different classes C_i is a constant given the input.

[3]We don't need to use them anymore, even for subsequent small items because the used space that we can assure they contain is the same.

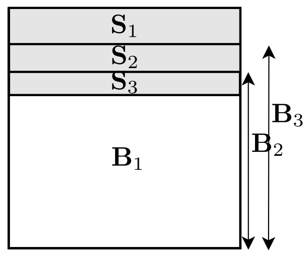

Figure 3: Configuration in which the bin has been closed because of a small C-item.

the execution of the algorithm by assigning new **S**-areas to **B**. In order to simplify the analysis of the different areas, the **B**-area is situated at the bottom of the bin, and the initial **S**-area at the top.

The contiguos space (not divided into smaller parts by the allocation of an **S**-area in it) of the **B**-area used to pack big items is named the *active part of the **B**-area*.

The area in which C-items are currently being packed is named the *active **S**-area*. All the other **S**-areas, if there are any, are named *closed **S**-areas*. Whenever a newly arrived C-item cannot be packed into the active **S**-area, this one is closed. Then there are two possibilities, if the allocation strategy for **S**-areas cannot allocate a new **S**-area then the bin is closed, and the C-item is packed in the new opened bin. Otherwise if there is space enough in the active part of the **B**-area, the new active **S**-area is allocated right under all the A-items of the active part of the **B**-area, and the remaining **B** space turns into the new active part of the **B**-area.

Figure 4: Example of allocation of **S**-areas and the equivalent used for the analysis of the used space.

The more **S**-areas allocated, the smaller their heights are, and the heights of them depend on the maximum number of **S**-areas possible as explained in section 2.3. The used case here is with a maximum of three **S**-areas. More areas cannot improve the asymptotic competitive ratio on the worst case scenario.

2.4 Packing Strategy Analysis

Let n be the number of bins used by the packing strategy, $\mathcal{B}_1$ the set of bins bin with one **S**-area closed and $\mathcal{B}_2$ similary with two **S**-areas. Let $\mathcal{B}_3$ be the set of bins with three **S**-areas and closed because of a big item and $\mathcal{S}$ when closed because of a small item. Let $\mathcal{B}$ be $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3$.

Let $\mathbf{S}_1$ be the initial **S**-area, and $\mathbf{S}_2$ and $\mathbf{S}_3$ the consecutives ones. $\mathbf{B}_1$ is the **B**-area in a bin with one **S**-area, and similary for $\mathbf{B}_2$ and $\mathbf{B}_3$. Note that a **B** area can have disjoint parts separated by **S**-areas.

Let $O^i_{\mathbf{S}_1}$, $O^i_{\mathbf{S}_2}$ and $O^i_{\mathbf{S}_3}$ be the occupied spaces of the i-th bin for the $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$ respectively. $O^i_{\mathbf{B}_1}$, $O^i_{\mathbf{B}_2}$ and $O^i_{\mathbf{B}_3}$ are defined the same way. If a bin does not have one kind of area (All the bins have exactly one kind of **B**-area, and at least one kind of **S**-area), the occupied space of that area is zero.

Note that the sets are disjoint as they represent different ways to close a bin and $|\mathcal{B}_1| + |\mathcal{B}_2| + |\mathcal{B}_3| + |\mathcal{S}| + 1 = n$. When n is big we can regard the number of bins as $|\mathcal{B}_1| + |\mathcal{B}_2| + |\mathcal{B}_3| + |\mathcal{S}|$ to calculate the average occupation for the worst case scenario on every set of bins:

$$\geq \frac{\sum_{i\in\mathcal{B}_1}(O^i_{\mathbf{B}_1}) + \sum_{i\in\mathcal{B}_2}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{B}_2}) + \sum_{i\in\mathcal{B}_3}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{B}_3}) + \sum_{i\in\mathcal{S}}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{S}_3})}{|\mathcal{B}_1| + |\mathcal{B}_2| + |\mathcal{B}_3| + |\mathcal{S}|}$$

$$\geq \min\{\frac{\sum_{i\in\mathcal{B}_1}(O^i_{\mathbf{B}_1})}{|\mathcal{B}_1|}, \frac{\sum_{i\in\mathcal{B}_2}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{B}_2})}{|\mathcal{B}_2|}, \frac{\sum_{i\in\mathcal{B}_3}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{B}_3})}{|\mathcal{B}_3|}, \frac{\sum_{i\in\mathcal{S}}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{S}_3})}{|\mathcal{S}|}\} \quad (1)$$

The area **B** can be considered as a contiguous area. Thus, its average occupied space can be easily analysed.

Lemma 1. *The average occupied space of every kind of* $\boldsymbol{B}$ *for bins* $\in \mathcal{B} \geq b/4 - 1/32$. *With* $b = height(\boldsymbol{B})$.

Proof. Let s_1, s_2 and s_3 be the $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$-area heights respectively. It can be assumed that the total height of the A-items is y and the height of the left and right side of B-items are y_1 and y_2 respectively, as in figure 1, and w.l.o.g., $y_1 \geq y_2$. It can also w.l.o.g be assumed that the bin is closed because of a big (A- and B- items), this will be demonstrated at the end.

W.l.o.g let the i-th bin be in $\mathcal{B}_j$, p^i_A and q^i_A be half of the occupied space of A-items in the i-th bin, i.e., with an A-item with height h in a bin, p^i_A and q^i_A are at least $h/4$ (A-items have width $\geq 1/2$). Let q^i_B be half of the occupied space of the side (left or right) with more occupied space, and p^i_B be the remaining occupied space of B-items in **B**.

For instance, in figure 1 the occupied space on the left side is bigger than the occupied space on the right side, so $q^i_B \geq y_1/12$ (B-item have width $\geq 1/6$). Note that the side with larger height may not be the one with a bigger occupied space.

For bins $\in \mathcal{B}$ the space occupied by C-items can be disregarded, as the closing bin configuration with less space would have none of them. The average occupied space of A- and B-items for bins $\in \mathcal{B}_j$ is:

$$\frac{\sum_{i=1}^{n}(p_A^i + q_A^i + p_B^i + q_B^i)}{|\mathcal{B}_j|} \geq \min_{i\in\mathcal{B}_j}\{p_A^i + q_A^{i+1} + p_B^i + q_B^{i+1}\}$$

Assume that the i-th bin belongs to $\mathcal{B}_j$, and that $p_A^i + q_A^{i+1} + p_B^i + q_B^{i+1} = \min_{k\in\mathcal{B}}\{p_A^k + q_A^{k+1} + p_B^k + q_B^{k+1}\}$. When a newly arrived item a cannot be packed in $\mathbf{B}$, a new bin has to be opened. The item a can be an A-item or a B-item, and the occupied space by B-items on the left side can be bigger or not than on the right side. Denote O_L for the occupied space on the left part by B-items and O_R for the right side, the average occupied space is:

- item a is an A-item.

  According to the packing strategy, a must be packed into the $(i+1)$-th bin. Asume y' is the height of a. By definition $p_A^i \geq y/4$.

  - $O_L > O_R$
    $p_B^i = O_L/2 + O_R \geq y_2/12 + (y_1 - y_2)^2/2 + y_2/6 = y_2/4 + (y_1 - y_2)^2/2$
    Because $O_L \geq y_2/6 + (y_1 - y_2)^2$
  - $O_L \geq O_R$
    $p_B^i = O_L + O_R/2 \geq O_L + O_L/2 \geq y_2/6 + y_2/12 + (y_1 - y_2)^2/2 = y_2/4 + (y_1 - y_2)^2/2$
    Because $O_R > O_L \geq y_1/6 \geq y_2/6$

  Thus, no matter which side is larger, $p_B^i \geq y_2/4 + (y_1 - y_2)^2/2$

  Combine with $q_A^{i+1} \geq y'/4$ from item a,

$$\begin{aligned}
&p_A^i + p_B^i + q_A^{i+1} + q_B^{i+1}\\
=&p_A^i + p_B^i + q_A^{i+1}\\
\geq&y/4 + [y_2/4 + (y_1 - y_2)^2/2] + y'/4\\
=&(y + y' + y_2)/4 + (y_1 - y_2)^2/2\\
\geq&(b - y_1 + y_2)/4 + (y_1 - y_2)^2/2\\
=&b/4 + (y_1 - y_2 - 1/4)^2/2 - 1/32\\
\geq&b/4 - 1/32
\end{aligned}$$

- item a is a B-item.

  The width of $a \geq 1/6$. Asuming y' is the height of a. The occupied space of a is at least y'^2 because items are defined so that their height is no bigger than their width, by definition $p_A^i \geq y/4$.

  - $O_L \geq O_R$. In this case $y + y_2 + y' > u$.
    $p_B^i = O_L/2 + O_R \geq y_1/12 + y_2/6$

– $O_L < O_R$. In this case $y + y_1 + y' > u$.
$p_B^i = O_L + O_R/2 \geq O_L + O_L/2 \geq y_1/6 + y_1/12 \geq y_2/6 + y_1/12$

Thus, no matter which side is larger, $p_B^i \geq y_1/12 + y_2/6$.

Combine with $q_B^{i+1} \geq y'^2/2$ from item a (since q_B^{i+1} is equal to half of the occupied space of the larger side),

$$
\begin{aligned}
& p_A^i + p_B^i + q_A^{i+1} + q_B^{i+1} \\
=& p_A^i + p_B^i + q_B^{i+1} \\
\geq& y/4 + [y_1/12 + y_2/6] + y'^2/2 \\
\geq& y/4 + [y_1/12 + y_2/6] + (b - y - y_2)^2/2 \\
\geq& (y/4 + y_2/4) + (b - y - y_2)^2/2 \\
=& (b - y - y_2 - 1/4)^2/2 + (b/4 - 1/32) \\
\geq& b/4 - 1/32
\end{aligned}
$$

Let's demonstrate now that w.l.o.g. the bin is closed by a big item. This is always the case for $\mathcal{B}_3$ by definition. For $\mathcal{B}_1$ if the bin is closed because of a small item, $y_1 \geq 1 - s_1 - s_2 \geq 1 - 2s_1$, and the occupied space for big items on the worst case is bigger than $2(1-2s_1)/6-1/36$, and for small items $s_1/3-1/36$ (using lemma 2). This space is bigger than the one closing the bin because of a big item, $(1 - s_1)/4 - 1/32$ (using lemma 1) for s_1 smaller than $17/24 \approx 0.7$, which is much bigger than the s_1 used for the algorithm, and so it holds.

For $\mathcal{B}_2$ if the bin is closed because of a small item, $y_1 \geq 1 - s_1 - s_2 - s_3 \geq 1 - s_1 - 2s_2$ and the occupied space for big items on the worst is bigger than $2(1-s_1-2s_2)/6-1/36$, and for small items $s_1/3-1/36+s_2/3-1/36$. This space is bigger than the one closing the bin because of a big item, $(1-s_1-s_2)/4-1/32$, because solving the inequality resulting of balancing the space when the bin is closed because of a small item and because of a big item we get:

$$(1 - s_1 - s_2)/4 - 1/32 \overset{?}{<} [(1 - s_1 - 2s_2)/3 - 1/36] + [s_1/3 - 1/36 + s_2/3 - 1/36]$$

$$8s_2 \overset{?}{<} 24s_1 + 3$$

This holds because $s_2 < s_1$. So we can asume that the bin is closed because of a big item w.l.o.g.

Thus, the lemma is correct.

□

Lemma 2. *The average occupied space of every kind of* $\boldsymbol{S}$ *closed* $\geq s/3 - 1/36$. *With* $s = height(\boldsymbol{S})$.

Proof. For a fixed $\mathbf{S}_j$, define $\overline{col_i}$ for every C_i as the set of all the columns of size w_i but col_i. Informally, $\overline{col_i}$ are the closed columns that contain elements of the class C_i and these are almost full. Let p_i^j and q_i^j be the disjoint occupations in the j-th column for the subclass C_i. Define p_i^j equal to $w_i/2 \cdot (l - h_j^i)$ with h_j^i being the free space at the top of the j-th column for the subclass C_i, and q_i^j as the remaining occupied space in the same column.

The total occupied space in $\mathbf{S}_j$ is the occupied space of every column of every subclass C_i.

$$\sum_i \sum_{1 \leq j \leq |\overline{col_i}|+1} (p_i^j + q_i^j) \geq \sum_i \min_{1 \leq j \leq |\overline{col_i}|} \{p_i^j + q_i^{j+i}\} \cdot |\overline{col_i}| \qquad (2)$$

W.l.o.g. assume $p_i^j + q_i^{j+i} = \min_{1 \leq j \leq |\overline{col_i}|} \{p_i^j + q_i^{j+i}\}$. Because the j-th column of the subclass C_i with free height h_j is a column of $\overline{col_i}$, cannot satisfy the newly arrived item a of size (x, y) from the class C_i. Note that $w_i/2 < x \leq w_i$ and $y > h_j^i$. Note also that $x > h_j$, because $x \geq y$, i.e. the width of an item is no less than its height.

The different cases are:

- $h_j \leq w_i/2$
  $p_i^j + q_i^{j+1} \geq p_i^j = w_i/2 \cdot (s - h_j^i) \geq w_i/2 \cdot (s - w_i/2) \geq w_i/2 \cdot (s - 1/12)$
- $h_j = w_i/2 + z > w_i/2$
  The item a can be partitioned into two disjoint parts of sizes $w_i/2 \cdot y$ and $(x - w_i/2) \cdot y$ which belong to p_i^{j+1} and q_i^{j+1} respectively. Thus, $q_i^{j+1} \geq (x - w_i/2) \cdot y$ and we have:

$$\begin{aligned}
& p_i^j + q_i^{j+1} \\
\geq & w_i/2 \cdot (s - h_j) + (x - w_i/2) \cdot y \\
\geq & w_i/2 \cdot (s - w_i/2 - z) + (y - w_i/2) \cdot y \\
\geq & w_i/2 \cdot (s - w_i/2 - z) + (h_j^i - w_i/2) \cdot y \\
= & w_i/2 \cdot (s - w_i/2 - z) + z \cdot w_i/2 \\
= & w_i/2 \cdot (s - w_i/2) \\
\geq & w_i/2 \cdot (s - 1/12)
\end{aligned}$$

Thus, no matter how large the free height on the top of the j-th column of the class C_i is, $p_i^j + q_i^{j+1} \geq w_i/2 \cdot (s - 1/12)$.

From the inequality 2 and the above analysis, it can be infered that the average occupied space for any kind of $\mathbf{S}$ closed is

$$\geq \sum_i (w_i/2 \cdot (s - 1/12)) \cdot |\overline{col_i}| = (s - 1/12)/2 \sum_i w_i \cdot |\overline{col_i}|$$

When the newly arrived item a cannot be packed, one of the active columns has to be closed, and so there is one subclass C_i without col_i. Since the width of the free space f is less than w_i (or else another column would be allocated and labeled active) and $\sum 1/6 + 1/12 + \cdots = 1/3$, can be infered than the sum of the width of the active column for all subclasses is $\leq 1/3$. Thus, the sum of the width of the almost full $\overline{col}$ is $\geq 2/3$, and it can be concluded that the average occupied space of every kind of closed **S** is:

$$\geq \frac{s - 1/12}{2} \cdot \frac{2}{3} = \frac{s}{3} - \frac{1}{36}$$

□

2.5 Allocation of S-areas Analysis and Upper Bound

As can be seen in the equation 1, the average occupation can be computer as a minimum.

$$\geq \min\{\frac{\sum_{i \in \mathcal{B}_1}(O^i_{\mathbf{B}_1})}{\mathcal{B}_1}, \frac{\sum_{i \in \mathcal{B}_2}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{B}_2})}{\mathcal{B}_2}, \frac{\sum_{i \in \mathcal{B}_3}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{B}_3})}{\mathcal{B}_3}, \frac{\sum_{i \in \mathcal{S}}(O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{S}_3})}{\mathcal{S}}\}$$

This means that in order to find the optimal sizes for the allocated **S**-areas, so that the bin has the biggest occupied space on the worst case, we need to balance the occupied spaces in the different possible bin closing configurations.

Let $O_{\mathcal{B}_1}$ be the minimum occupied space needed for a bin $\in \mathcal{B}_1$ to close with a possible newly arrived item. Define $O_{\mathcal{B}_2}$, $O_{\mathcal{B}_3}$ and $O_{\mathcal{S}}$ the same way.

$$O_{\mathcal{B}_1} = O_{\mathcal{B}_2} = O_{\mathcal{B}_3} = O_{\mathcal{S}} \tag{3}$$

$$\begin{aligned} O_{\mathcal{B}_1} &= O^i_{\mathbf{B}_1} \\ O_{\mathcal{B}_2} &= O^i_{\mathbf{S}_1} + O^i_{\mathbf{B}_2} \\ O_{\mathcal{B}_3} &= O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{B}_3} \\ O_{\mathcal{S}} &= O^i_{\mathbf{S}_1} + O^i_{\mathbf{S}_2} + O^i_{\mathbf{S}_3} \end{aligned}$$

Let s_1, s_2 and s_3 be the $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$-area heights respectively. The system of equations 3 has the solution $s_1 = 1103/4200 \approx 0.26$, $s_2 = 251/1050 \approx 0.24$ and $s_3 = 109/525 \approx 0.21$, by using the lemmas 1, $O_{\mathcal{S}} \geq s/3 - 1/36$ and 2, $O_{\mathcal{B}} \geq b/4 - 1/32$. This way the balanced occupied space is 643/4200, which means that the average occupation in each bin is at least 0.153. Thus, this strategy is 6.53-competitive.

For the worst case analysis of this strategy, consider the packing in the **B**-area of a sequence of items $(b, a, b, a, b, a, \cdots)$ that arrive one at a time. $a \in A$, and $b \in B$. The size of b is 0.25×0.25, while the size of a is $0.5 \times (\frac{4200-1103}{4200} - 0.25 + \epsilon)$, where ϵ is a very small value. According to the strategy, a and b cannot be packed into the same bin. Thus, the amortized occupied space in each bin is $(0.25^2 + 0.5(\frac{4200-1103}{4200} - 0.25 + \epsilon))/2 = 0.25\epsilon + 643/4200$. Therefore, the strategy is tight.

3 A new Lower Bound

In this section, we will show that the lower bound of the competitive ratio for any 1-bounded space algorithm is $2.\hat{6}$, which is an improvement over the previous bound of 2.5. The input used to show such a lower bound is a variation based in the one used by Chin. et al. [1] to reach the lower bound of 2.5. The key point in this variation is the use of a padding item P, its main characteristic is that all the P items together can fit in at most one square bin.

Consider the following sequence of rectangular items as the input:

$$(A_1, B_1, A_2, B_1, \cdots A_{2k+2}, B_1, X_1, X_2, X_3, k/3 \cdot (3 \cdot B_2, 3 \cdot C, P))$$

Where A_i is $(1/3 + a_i, 2/3 + x)$, B_1 is $(1/3 - \epsilon, 1/3 - \epsilon)$, B_2 is $(1/3 + 2 \cdot \epsilon, 1/3 + 2 \cdot \epsilon)$, P is $(1, \delta)$ and C is $(2/3 + \epsilon, 1/3 - x)$. The sizes of X_1, X_2 and X_3 are shown in figure 5 and their sizes are so that after adding them the square bin is completly full.

The sizes of these rectangular items must satisfy the constrains:

$$\begin{aligned}
&a_1 = \epsilon_a, a_{2i} = (i+1)\epsilon_a, \text{ and } a_{2i+1} = -i\epsilon_a - \delta_a \text{ for } i > 0,\\
&\delta_a > 2\epsilon > x > \epsilon > 0,\\
&\epsilon_a > \delta_a + \epsilon,\\
&\delta > 2x - 2\epsilon,\\
&a_i, \delta_a, \epsilon, \epsilon_a, x, \delta \gg 1
\end{aligned}$$

Lemma 3. *For any online packing strategy, the number of used bins for the input sequence is at least* $8k/3 + 2$.

Proof. The number of bins used is analysed for differents parts of the input sequence.

$$(\overbrace{A_1, B_1, A_2, B_1, \cdots A_{2k+2}, B_1, X_1, X_2, X_3}^{2k+2 \text{ bins}}, \overbrace{k/3 \cdot (\overbrace{3 \cdot B_2}^{1 \text{ bin}}, \overbrace{3 \cdot C, P}^{1 \text{ bin}}))}^{2k/3 \text{ bins}}$$

- As can be seen in figure 5 , the items A_i, B_1 and A_{i+1} cannot be put in the same bin, because B_1 cannot be packed together with A_i and A_{i+1} because $(2/3 + x) + (1/3 - \epsilon) > 1$, and the sum of the heights is $(2/3 + \epsilon_a - \delta_a) + (1/3 - \epsilon) > 1$ for $i \geq 2$ [4]. Thus, the first $4k + 4$ rectangular items must use at least $\mathbf{2k + 2}$ **bins**. Items X_1, X_2 and X_3 can be packed into the $2k + 2$-th bin, thus, to pack the first item B_2 a new bin must be opened.

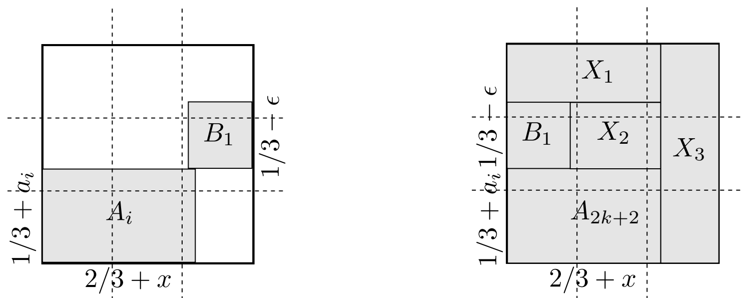

Figure 5: A_i and A_{i+1} cannot be packed into the same bin.

- The best packing for the following $k/3$ groups of items $(3 \cdot B_2, 3 \cdot C, P)$ uses two bins per group.
  - The three B_2 items of the group can be packed into one bin, although only two B_2 items per row and per column can fit because $3 \cdot (1/3 + 2\epsilon) > 1$. This means that the following C item cannot fit into the same bin and another bin has to be opened because $(2/3 + 4\epsilon) + (1/3 - x) > 1$ and $(1/3 + 2\epsilon) + (2/3 + \epsilon) > 1$, as can be seen in figure 6.
  - The three C items and the padding P item of the group can be packed in one bin. If the three C items are packed contiguously and with the same orientation, then the following B_2 item must be packed in a new bin because $(1/3 + 2\epsilon) + (2/3 + \epsilon) > 1$, otherwise, if one of the C items is perpendicular to the other two, the padding P item ensures that the following B_2 item has to be packed into a new bin, this is so because $(2/3 - 2x) + (\delta) + (1/3 + 2\epsilon) > 1$ as can be seen in figure 6.

  Thus, 2 bins must be used to pack a group. Having $k/3$ groups, a total of $\mathbf{2k/3}$ **bins** are needed.

In total there are $\mathbf{8k/3 + 2}$ **bin**.

□

[4]For the special case of $i = 1$ the sum of the heights is $(2/3 + 3\epsilon_a) + (1/3 - \epsilon) > 1$

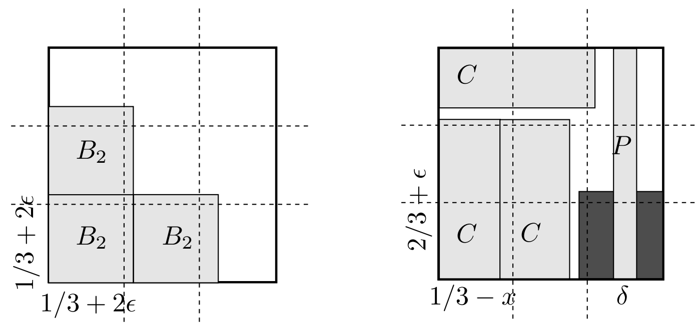

Figure 6: Restrictions caused by padding P

Lemma 4. *The optimal solution for packing the above input uses* $k + 4$.

Proof. The different agrupations used to pack the items are defined below.

- Define $\delta \leq 3/k$ so that all the P items can be packed in **1 bin**.
- The items A_1, A_3, $2 \cdot B_1$, B_2 and C can be packed together in **1 bin** as shown in figure 7, because the sum of the heights of A_1, A_3 and B_2 is $1 - \delta_a + 2\epsilon < 1$.
- The items A_{2i}, A_{2i+3}, $2 \cdot B1$, $B2$ and C can as well be packed together for $0 < i < k$, for the same reason. They can be packed using a total of $\mathbf{k - 1}$ **bins**.
- The items A_{2k}, A_{2k+2} that remain cannot be packed in the above way (Note that the above packing is possible because the sum $a_{2i} + a_{2i+3} = -\delta_a$), these two items and the remaining items X_1, X_2 and X_3 can be packed using at most 3 **bins**.

  Thus, the optimal offline algorithm uses at most $\mathbf{k + 4}$ **bins** for the given input.

□

Theorem 1. *The lower bound of competitive ratio for 1-bounded space online algorithm is* $2.\hat{6}$

Proof. From 3 and 4 we can reach this lower bound because:

$$O(\frac{8k/3 + 2}{k + 4}) = \frac{8}{3} = 2.\hat{6}$$

□

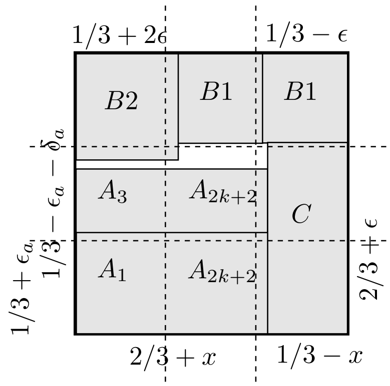

Figure 7: Packing of all the items but the padding item P in one bin by the offline packing strategy.

References

[1] Francis Y.L. Chin, Hing-Fung Ting and Yong Zhang, *1-Bounded Space Algorithms for 2-Dimensional Bin Packing.* Algorithms and Computation, Department of Computer Science, The University of Hong Kong, Hong Kong

[2] J Csirik, DS Johnson, *Bounded space on-line bin packing: best is better than first.* Algorithmica, 2001 - Springer

[3] C. C. Lee, D. T. Lee, *A simple on-line bin-packing algorithm.* J. ACM, Vol. 32, No. 3. (1985), pp. 562-572

[4] E. G. Coffman, Jr. , M. R. Garey , D. S. Johnson, *Approximation algorithms for bin packing: a survey.* Approximation algorithms for NP-hard problems, PWS Publishing Co., Boston, MA, 1996

[5] S. Fujita and T. Hada, *Two-Dimensional On-Line Bin Packing Problem with Rotatable Items.* Proc. of COCOON, LNCS 1858, 210220, 2000

[6] S. Fujita, *On-Line Grid-Packing with a Single Active Grid.* Ibarra, O.H., Zhang, L. (eds.) COCOON 2002. LNCS, vol. 2387, pp. 476483. Springer, Heidelberg (2002)