A Tale of Two Dimensional Bin Packing

Nikhil Bansal
IBM T.J. Watson,
Yorktown, NY 10598
nikhil@us.ibm.com

Andrea Lodi
DEIS, Univ. of Bologna, Italy
and IBM T.J. Watson,
Yorktown, NY 10598
alodi@us.ibm.com

Maxim Sviridenko
IBM T.J. Watson,
Yorktown, NY 10598
sviri@us.ibm.com

Abstract

The 2-dimensional Bin Packing problem (2BP) is a generalization of the classical Bin Packing problem and is defined as follows: Given a collection of rectangles specified by their width and height, pack these into the minimum number of square bins of unit size. We study the case of 'orthogonal packing without rotations', where rectangles cannot be rotated and must be packed parallel to the edges of a bin.

Often in practical cases of 2BP problems there are additional constraints on how complicated the packing patterns in a bin can be. A well-studied and frequently used constraint is that every rectangle in the packing must be obtainable by recursively applying a sequence of edge-to-edge cuts parallel to the edges of the bin. Such cuts are known as guillotine cuts.

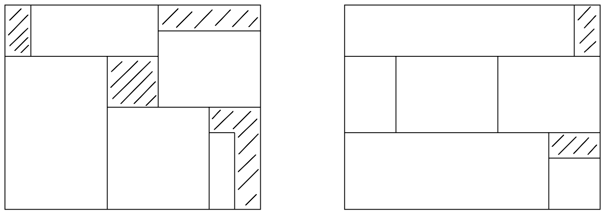
Our main result is that the guillotine 2BP problem admits an asymptotic polynomial time approximation scheme. This is in sharp contrast with the fact that the general 2BP problem is APX-Hard. En route to our main result, we show a structural theorem about approximating general guillotine packings by simpler packings, which could be of independent interest.

1. Introduction

In the *two-Dimensional Bin Packing Problem* (2BP) we are given a collection of rectangles specified by their width and height, that have to be packed into larger containers (*bins*). The most interesting and well-studied version of this problem is the so-called *orthogonal packing without rotation* where each rectangle must be packed parallel to the edges of a bin, and no rotations are allowed. The goal is to find a feasible packing, i.e., a packing where rectangles do not overlap, using the minimum number of bins.

In many practical cases of two-dimensional packing problems there are additional constraints on the patterns that can be used to pack items in a bin. One of the well-studied constraints that occurs often in practice is that every rectangle can be obtained by applying recursively a sequence of edge-to-edge cuts parallel to the edges of the bin. Such cuts are known as guillotine cuts. Figure 1 depicts both non-guillotine (Figure 1(i)) and guillotine (Figure 1(ii)) bin patterns. The guillotine constraint arises naturally from the design of cutting machines and the complexity of programming them. It is mainly relevant in cases where the raw material to be cut has a low cost with respect to the industrial costs involved in the cutting process. This is particularly true in many wood, paper, glass or rubber cutting applications. For example, real-world guillotine problems have been studied by Schneider [17] in a crepe-rubber mill, by Puchinger et al. [16] in the glass industry, and by Mc Hale and Shah [15] in the paper cutting context. Infact, a Google search with the keywords "guillotine","cutting" and "optimization" gives links to hundreds of companies that sell optimization software for finding good cutting layouts. See for example http://www.proas.hr/2DOptimization/, http://www.tmachines.com/ and http://www.optimizecutter.com/. Another reason for considering guillotine packing is that one can design very effective heuristics based on column generation techniques (see, Gilmore and Gomory [11]) as done for example by de Carvalho and Rodrigues [9] and by Vanderbeck [18].

Aside from the practical interest, guillotine packings have been studied extensively from a theoretical viewpoint. Gilmore and Gomory [12] initiated the systematic study of guillotine packings by introducing the notion of *k-stage packing*. Here each stage consists of either horizontal or vertical guillotine cuts (but not both). On each stage each of the sub-bins obtained



(i) (ii)

Figure 1. (i) non-guillotine and (ii) guillotine patterns. The shaded area indicates free unused space.

on the previous stage is considered separately and can be cut again by using horizontal or vertical guillotine cuts. The number of cuts used to obtain each rectangle is at most k, plus an additional cut to separate the rectangle itself from a waste area. For example, the packing in Figure 1(ii) is a 2-stage packing. These k-stage packing can have a very rich structure even for small values such as $k = 2$ and $k = 3$. Infact, the best known approximation algorithms for the general two-dimensional bin packing problem due to Caprara [3] and for the related two-dimensional strip packing problem (a strip is a container of unit width and infinite height and the problem goal is to find a feasible packing in the strip which minimizes the height) used due to Kenyon and Rémila [14], produce packing that are 2-stage and 3-stage respectively. The $k = 2$ case in particular has been extensively studied and is usually referred to as *shelf* packing in the literature. (We refer the reader to Coffman et al. [6] for more details on approximation algorithms for bin packing).

In this paper, we give an approximation algorithm for the general guillotine packing problem. We use the notion of asymptotic approximation ratio (which is standard for bin-packing algorithms) to measure our performance. An algorithm A for a minimization problem has asymptotic approximation ratio ρ if $A(I) \leq \rho \cdot OPT(I) + C$ for all problem instances I, where $A(I)$ (resp. $OPT(I)$) denotes the value of algorithm A (resp. the optimum value) on the instance I and C is a constant independent of I. Similarly an *Asymptotic Polynomial Time Approximation Scheme* (APTAS) is an algorithm that receives as input a required *accuracy* $\varepsilon > 0$, runs in time polynomial in the size of I, and produces a solution of value at most $(1+\varepsilon) \cdot OPT(I) + C$. Here C is a constant independent of I but may depend on ε.

1.1. Previous Work

The classical one-dimensional bin packing, which is a special case of our problem, has been studied extensively since the 60's (see [6] and references therein). An important breakthrough was made by Fernandez de La Vega and Lueker [10] who gave the first APTAS for the problem. This was improved significantly later by Karmarkar and Karp [13].

Various approximation guarantees for two and higher dimensional versions of bin packing have been given in a long sequence of papers (see [3, 2, 5, 6] and references therein). The current best known asymptotic approximation guarantee for the 2BP problem is $T_\infty = 1.691\ldots$ due to Caprara [3]. As mentioned above, the algorithm of [3] infact produces a 2-stage (shelf) packing. Recently, Bansal et al. [2] showed that the general 2BP problem does not admit an APTAS.

For k-stage packing, Csirik and Woeginger [8] showed that increasing the number of allowed stages can help significantly. In particular, while general strip packing can be approximated arbitrary well by 3-stage packings [14], the asymptotic worst-case ratio between the optimal 2-stage and general strip packing is T_∞ [8]. Finally, for the case of 2-stage 2BP an APTAS was given recently by Caprara, Lodi and Monaci [4].

1.2. Our Results

Our main result is an asymptotic polynomial time approximation scheme for the guillotine 2BP problem. This result is proved in two parts: We first prove a structural theorem which shows that an arbitrary guillotine packing can be approximated arbitrary well by a packing that has a constant number of stages only. Second, we give an algorithm that takes an instance I and a parameter ε and produces a $(k+2)$-stage packing using $(1+\varepsilon) \cdot OPT_k(I) + O(1)$ bins, where $OPT_k(I)$ is the number of bins required by the optimum k stage packing for I.

1.3. Overview of Techniques

The guillotine subdivisions of a bin can be viewed naturally as a tree, where each node corresponds to a rectangular region, and given a node, its children at the next level are obtained by applying either vertical or horizontal cuts. A k-stage packing corresponds to a tree of depth k.

In the structural theorem we show how to modify a guillotine tree with an arbitrary depth into one with constant depth by repacking certain rectangles in a way that allows us to delete certain edges of the tree. It would be useful to consider Figure 2 which shows a guillotine packing of a bin which requires a non-constant number of stages. In fact it is easy to see that the set of items on the picture cannot be packed into a bin in any other way.

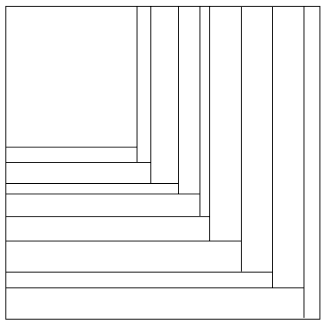


Figure 2. Guillotine packing with large number of stages.

Observe that if a packing has a lot of stages, then most sub-bins will contain a lot of "thin" or "short" rectangles only. We show that the heights and widths of these thin or short rectangles can be rounded to a constant number of types (without increasing the bins required in the optimum solution significantly). This allows us to argue that if a tree has a long path, then lot of edges on this tree are "similar" and can be merged in a way that reduces the depth of the tree.

Our algorithm for the k-stage guillotine bin packing problem for constant k uses a novel rounding technique. Previously, all known approximation schemes for bin packing problems rely on the fact that there is a total order on the items to be packed (or example in bin packing [10, 13] or squares packing [2], items can be ordered by size). This ordering is crucial for designing approximation schemes as it allows us to round the items and simplify the instance. However, for the general rectangle packing all rectangles could be incomparable and hence have no order among them. For example, consider the case when all rectangles in the input instance are such that their height and width sum to 1. Infact, this observation about incomparability of rectangles was critically used to show the APX hardness of the general 2BP problem [2]. Surprisingly, for our problem we can define a rounding which rounds widths and heights (of possibly incomparable items) up to widths and heights that were not present originally in the instance and prove that there is an optimal solution for the rounded instance of good quality. This rounding infact depends on certain structural properties of constant stage guillotine packings. We believe that our new rounding technique could be useful for other multi-dimensional packing problems that have resisted attempts to design approximation schemes for them.

2. Preliminaries

2.1. Definitions

Let $R = \{r_1, \ldots, r_n\}$ be a set of rectangles. Let w_r be the width and h_r be the height of a rectangle $r \in R$.

Definition 1 *The packing of rectangles into bins is called guillotine if each rectangle (item) can be obtained through a sequence of edge-to-edge cuts parallel to the edges of the bin where the item was packed.*

Every guillotine subdivision of a bin consists of *stages* where each stage consists of a set of parallel horizontal or vertical lines.

Definition 2 *The guillotine tree is a graph where each vertex corresponds to a rectangular region (sub-bin) obtained on a certain stage of guillotine subdivision of a bin. Two vertices are connected by an edge if a rectangular region corresponding to one vertex was obtained from a rectangular region corresponding to another vertex by one stage of guillotine subdivision. The root of the guillotine tree corresponds to a bin and every leaf corresponds to an individual rectangle plus probably some waste space.*

Definition 3 *A packing of rectangles into bins is called k-stage guillotine packing if k is a minimum number such that all paths in the guillotine tree corresponding to that packing from the root to some leaf consist of at most $k+1$ vertices.*

We will repeatedly use the classic Next Fit Decreasing Height (NFDH) algorithm to pack certain items. Due to space constraints, we refer the reader to [7] for a description of NFDH. In particular, NFDH produces a two-stage guillotine packing and satisfies the following properties:

Lemma 1 *[7] Let R be a set of rectangles with width and height values at most δ. If NFDH cannot place any other item in a rectangular bin of size $a \times b$ (with $a, b \leq 1$), then the total wasted (unfilled) area in that bin is at most $\delta(a+b)$.*

Lemma 2 *[7] Let R be an arbitrary set of rectangles with total area A. The packing found by NFDH uses at most $4A + O(1)$ unit size bins.*

3. Structural Theorem

Let $\varepsilon > 0$ be our precision parameter such that $1/\varepsilon$ is an integer. We define a sequence of numbers δ_i, $i = 0, \ldots, 1/\varepsilon$ iteratively $\delta_0 = 1$, and $\delta_{i+1} = \frac{\varepsilon \cdot \delta_i}{f(\delta_i)}$ where

$$f(x) = (1/x)^{g(x)} \cdot g(x) \text{ and } g(x) = 2\left(\frac{1}{x} + \left(\frac{1}{\varepsilon x}\right)^{1/x}\right).$$

Theorem 1 *Let $R = \{r_1, \ldots, r_n\}$ be a set of rectangles. If there is a guillotine packing of R into OPT bins then there is a $g(\delta_{1/\varepsilon})$-stage guillotine packing of R using $(1 + O(\varepsilon))OPT + O(1)$ bins.*

Proof: The general idea for the proof is to start with guillotine packing of rectangles in R and transform it into another guillotine packing that has constant number of stages. Such a transformation will be done in such a way that adds at most $O(\varepsilon)OPT + O(1)$ new bins to the existing solution.

3.1. Classification of Items

First we classify all rectangles in R according to their size. Define $U_i = \{r \in R | \delta_i < w_r \leq \delta_{i-1} \text{ or } \delta_i < h_r \leq \delta_{i-1}\}$, for $i = 1, \ldots, 1/\varepsilon$. Since there are $1/\varepsilon$ sets U_i and every rectangle appears in at most two of them there is at least one index $p \in \{1, \ldots, 1/\varepsilon\}$ such that the total area of rectangles in U_p is at most 2ε of the total area of the rectangles in the instance, i.e., $2\varepsilon \sum_{r \in R} w_r h_r$.

We call all rectangles in U_p *medium* and pack them separately by NFDH using $O(\varepsilon)OPT + O(1)$ bins by Lemma 2. The remaining rectangles are partitioned into *big*, *horizontal*, *vertical* and *small* and denoted by B, H, V and S respectively, as follows:

$$\begin{aligned}
B &= \{r \in R | w_r > \delta_{p-1} \quad \text{and} \quad h_r > \delta_{p-1}\},\\
H &= \{r \in R | w_r > \delta_{p-1} \quad \text{and} \quad h_r \leq \delta_p\},\\
V &= \{r \in R | w_r \leq \delta_p \quad \text{and} \quad h_r > \delta_{p-1}\},\\
S &= \{r \in R | w_r \leq \delta_p \quad \text{and} \quad h_r \leq \delta_p\}.
\end{aligned}$$

3.2. Fractional Guillotine Packing

Consider the instance $R' = R \setminus \{S\}$, (i.e., R restricted to rectangles in B,V and H only). We say that a packing of R' is *fractional*, if we are allowed to divide a rectangle in H arbitrarily into smaller rectangles of the same width, and a rectangle in V arbitrarily into smaller rectangles of same height and pack these subrectangles at different positions. The fractional packing is allowed to have at most n stages. Throughout this section we use OPT to denote the number of bins used by the optimum fractional guillotine packing of R'. Clearly, OPT is a lower bound on the optimum (integral) guillotine packing of R.

We will show how to obtain another guillotine fractional packing of R' that has at most $g(\delta_{1/\varepsilon})$ stages and uses at most $(1+O(\varepsilon))OPT + O(1)$ bins. We then give a way to obtain an integral guillotine packing from this fractional packing such that the number of stages increases by at most 1, and the number of additional bins required is $O(\varepsilon)OPT + O(1)$. Finally, we show how to add the rectangles in S to this packing of R', such that the overall packing is still guillotine with a constant number of stages and the number of bins required is essentially the same as used by the optimum guillotine packing for R.

3.3. Rounding

We first show how to round up the height of items in V and the width of items in H to a constant number of possible values without significantly increasing the number of bins needed. The rounding scheme is the one used first by Kenyon and Rémila [14] for strip packing and can be viewed as a fractional version of the scheme introduced by Fernandez de la Vega and Lueker [10] for one-dimensional bin packing.

We order all rectangles in V in decreasing order of their height, and let W_V denote the cumulative total width of the rectangles in V. We partition V into $x = 1/(\varepsilon \cdot \delta_{p-1})$ groups $G_1, G_2, \ldots, G_x$ consisting of consecutive rectangles in the height order such that the total width of rectangles in each group is exactly $\varepsilon \cdot \delta_{p-1} W_V$. Note that some rectangles may be split vertically between groups. For each rectangle, we now round up its height to that of the tallest rectangle in its group.

It remains to show that there is a fractional packing for this rounded instance that uses $(1 + O(\varepsilon)) \cdot OPT + O(1)$ bins. Consider the optimum fractional packing and delete rectangles from the group G_1 and pack them in separate bins using NFDH. For each $i \geq 2$, we now use the places originally used by rectangles G_{i-1} to place the rounded up rectangles in G_i. This can be done without conflict because the rounded up height of any rectangle in G_i does not exceed the original height of any rectangle in G_{i-1}, the total width of rectangles in each group is the same and we are allowed to pack rectangles in V fractionally.

The total area of the items in the group G_1 is at most $W_V/x = \varepsilon \cdot \delta_{p-1} W_V$, and since each item in V has height at least δ_{p-1} we also have that the total area of rectangles in V is at least $\delta_{p-1} W_V$ and hence $OPT \geq \delta_{p-1} W_V$. Thus, by Lemma 2 we need

$O(\varepsilon) \cdot OPT + O(1)$ additional bins to pack items from G_1. We may also assume that pieces corresponding to one rectangle have the same rounded height since there are at most x, i.e. a constant number, rectangles that do not satisfy that property and we could pack them separately using constant number of bins. We apply an analogous procedure for rectangles in H. Let $n_v = n_h = 1/(\varepsilon \cdot \delta_{p-1})$ denote the number of types for items in H and V.

3.4. Transformation of a guillotine tree

Consider the optimum fractional guillotine packing of rectangles from B and rounded rectangles from H and V. As in the integral case, we can view the fractional packing in each bin as a guillotine tree, where each leaf in this tree corresponds to a rectangular region that contains exactly one rectangle from B or one fractional piece of a rectangle from H or V plus some waste space. We now show how to transform this fractional guillotine packing with arbitrary number of stages into an integral guillotine packing with at most $g(\delta_{p-1}) \leq g(\delta_{1/\varepsilon})$ stages.

3.4.1 Labeling the Edges

Consider a guillotine tree T, and consider an edge (u, v) where u is a parent and v is a child. Let R_u and R_v be the rectangular regions in the guillotine subdivision of the bin corresponding to vertices u and v. By definition $R_v \subseteq R_u$. Assume that R_v was obtained from R_u by vertical cuts (the proof for horizontal cuts is identical, we just need to switch widths and heights in the argument below). Let C_u be the set of all children of u except vertex v in T. If the total width of all rectangular regions corresponding to vertices in C_u is at most δ_{p-1}, then we *label* the edge (u, v). Since for every unlabelled edge either the height or the width of a child decreases by at least δ_{p-1}, there can be at most $2/\delta_{p-1}$ of unlabelled edges in any root to leaf path P. Thus, it suffices to show how to transform the packing such that any path P has a constant number of labelled edges.

Observe that if edge (u, v) is labelled then the cumulative width of the rectangular regions corresponding to C_u is at most δ_{p-1} and hence they can contain only rectangles from V. We contract all the subtrees rooted at vertices in C_u to form one *supervertex* $\tilde{C}_u$ and define an edge $(u, \tilde{C}_u)$. The height of the region corresponding to $\tilde{C}_u$ is the height of R_u and its width is equal to the total width of regions corresponding to the vertices in C_u. Thus, R_u is subdivided into two rectangular regions R_v and $\tilde{C}_u$ by one vertical cut. We next give a procedure to repack the items in $\tilde{C}_u$ into a 2-stage guillotine packing. This new repacking of $\tilde{C}_u$ will be useful later when we modify the height of guillotine tree.

3.4.2 Repacking the Supervertices

We now repack the rectangles (or pieces of rectangles) in $\tilde{C}_u$. We refer the reader to Figure 3 to view this process. Given a packing in $\tilde{C}_u$ we extend the right and left edges of every rectangle or piece of a rectangle to define an edge-to-edge vertical cut of $\tilde{C}_u$. In this process some rectangles may be divided vertically into smaller rectangles, but this is allowed, as we are interested in a fractional packing only. Consider a vertical rectangular region between two consecutive vertical cuts. This contains a set of pieces of rectangles from V. We shift all the pieces in this rectangular region down such that there is no waste space between two pieces and there is no waste space between the bottom edge of $\tilde{C}_u$ and first piece in the rectangular region.

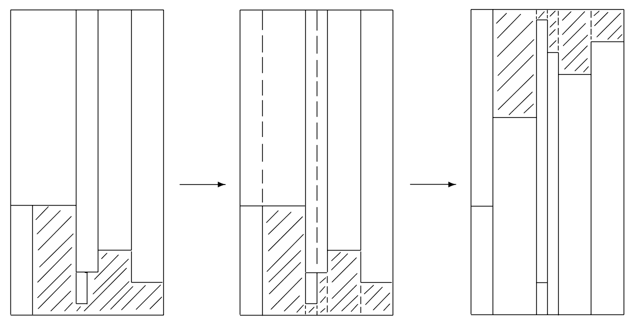


Figure 3. Repacking in supervertices of the guillotine tree.

Let $h(\tilde{C}_u)$ denote the height of the packing in $\tilde{C}_u$ after this repacking process. That is, $h(\tilde{C}_u)$ is the distance between the lower edge of the rectangular region corresponding to $\tilde{C}_u$ and the topmost edge among the rectangles packed in $\tilde{C}_u$. We claim that for any labelled edge (u, v), the value $h(\tilde{C}_u)$ can only range over a constant number of values. This follows as $h(\tilde{C}_u)$ is some integer linear combination of heights of rectangles in V. As each rectangle in V has height at least δ_{p-1}, $h(\tilde{C}_u)$ is a combination of at most $1/\delta_{p-1}$ terms. Finally, as there are at most $1/(\varepsilon \cdot \delta_{p-1})$ candidate heights in V, the number of candidates for $h(\tilde{C}_u)$ is at most $(\frac{1}{\varepsilon \cdot \delta_{p-1}})^{1/\delta_{p-1}}$.

3.4.3 Transforming the Tree

Suppose that the number of labelled edges in the path P, that correspond to stages with vertical cuts is larger than $(\frac{1}{\varepsilon \cdot \delta_{p-1}})^{1/\delta_{p-1}}$. Then, there are at least two labelled edges (u, v) and $(\bar{u}, \bar{v})$ corresponding to stages with vertical cuts such that $h(\tilde{C}_u) = h(\tilde{C}_{\bar{u}})$. This means that though the rectangular regions corresponding to $\tilde{C}_u$ and $\tilde{C}_{\bar{u}}$ have different heights, after the repacking process, the packing in each of these regions has the same height.

We now describe our transformation, which is perhaps the most critical part of the proof. Consider the guillotine tree, and suppose that (u, v) and $(\bar{u}, \bar{v})$ are two labelled edges for which $h(\tilde{C}_u) = h(\tilde{C}_{\bar{u}})$. For example see Figure 5. Without loss of generality, we assume that for any labelled edge (u, v), the supervertex $\tilde{C}_u$ is the right child of u. Let $w(\tilde{C}_u)$ and $w(\tilde{C}_{\bar{u}})$ be the widths of the rectangular region associated with $\tilde{C}_u$ and $\tilde{C}_{\bar{u}}$ respectively.

We now show how to modify the packing such that the edge (u, v) can be deleted in the tree. We will remove the rectangular region associated with $\tilde{C}_u$ and place the packing in $\tilde{C}_u$ adjacent to the packing in $\tilde{C}_{\bar{u}}$. Since the height of the packing in $\tilde{C}_u$ is identical to that in $\tilde{C}_{\bar{u}}$, it suffices to have a procedure where we can extend the width of the region corresponding to $\tilde{C}_{\bar{u}}$ by $w(\tilde{C}_u)$. To do this, consider the rightmost boundary of the rectangular region corresponding to $\tilde{C}_{\bar{u}}$, we extend it in both directions until it touches the top and bottom edges of the rectangular region corresponding to u. Let $\mathcal{Q}$ denote the set of rectangular regions in the tree whose interior intersects this line or whose right boundary touches this line. We extend the width of all the regions in $\mathcal{Q}$ by $w(\tilde{C}_u)$. See Figure 4 to view this process. Here $\mathcal{Q} = \{a, b, c, d\}$. This gives a new guillotine subdivision where $\tilde{C}_{\bar{u}}$ has width equal to $w(\tilde{C}_u) + w(\tilde{C}_{\bar{u}})$. Since the vertex u only has one child v in this modified guillotine tree, we can delete the edge (u, v).

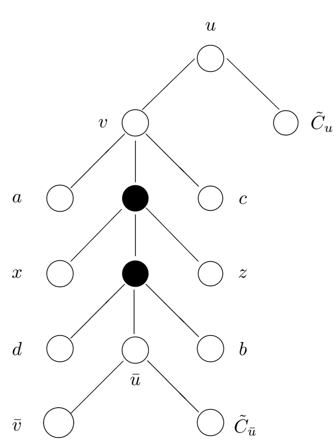


Figure 5. Guillotine tree for the packing.

Applying the above procedure repeatedly, we obtain a tree where in any root to leaf path, the number of labelled edges that correspond to stages with vertical cuts is at most $(\frac{1}{\varepsilon \cdot \delta_{p-1}})^{1/\delta_{p-1}}$. Similarly, the number of labelled edges corresponding to stages with horizontal cuts is also at most $(\frac{1}{\varepsilon \cdot \delta_{p-1}})^{1/\delta_{p-1}}$. Thus, the depth of the tree is at most

$$2/\delta_{p-1} + 2(\frac{1}{\varepsilon \cdot \delta_{p-1}})^{1/\delta_{p-1}} = g(\delta_{p-1}) \leq g(\delta_{1/\varepsilon}).$$

3.4.4 Modified Guillotine Trees

We further restrict the structure of guillotine trees in a rather straightforward way. Consider a vertex u and suppose that the children of u are obtained by vertical cuts. We merge all children of u with width smaller then δ_{p-1} to form the region $\tilde{C}_u$. As $\tilde{C}_u$ can only contain rectangles in V, we repack the rectangles in $\tilde{C}_u$ using the procedure described above which gives a 2-stage fractional packing. Applying this transformation (and an analogous one for horizontal cuts), the degree of u in this modified guillotine tree is at most $1/\delta_{p-1}$, and hence the tree has at most $(1/\delta_{p-1})^{g(\delta_{p-1})}$ leaves. This guillotine tree has few types of leaves. Either a leaf contains a single big item, or else it contains a 2-stage fractional packing of items in H or V. Note that a modified guillotine tree of depth k actually contains a $k+1$ stage packing, since the leaves in the last stage might actually have a two-stage packing of H or V.

3.5. Obtaining an Integral Packing

The final step is to show how to transform the fractional packing of H and V into an integral one (where the rectangles in H and V are not allowed to be split arbitrarily), and how to add the rectangles in S which we ignored thus far. This can be done using the (by now standard) technique introduced by Kenyon and Rémila [14] to convert a fractional solution to the strip packing problem into an integral one, and we only sketch the proof due to space constraints.

Compared to the setting of [14], the only difference is that while [14] had a single strip of unit width, we have multiple strips corresponding to each leaf containing items in H and V. However, the argument of [14] can be easily modified to work in our setting as the number of leaves in each tree is bounded by the constant $(1/\delta_{p-1})^{g(\delta_{p-1})}$ and as there are only a constant number of widths for rectangles in H and constant number of heights for rectangles in V, and hence a constant number of configurations.

Finally, adding the rectangles in S to the free spaces in leaves can exactly be done as in [14] and the analysis

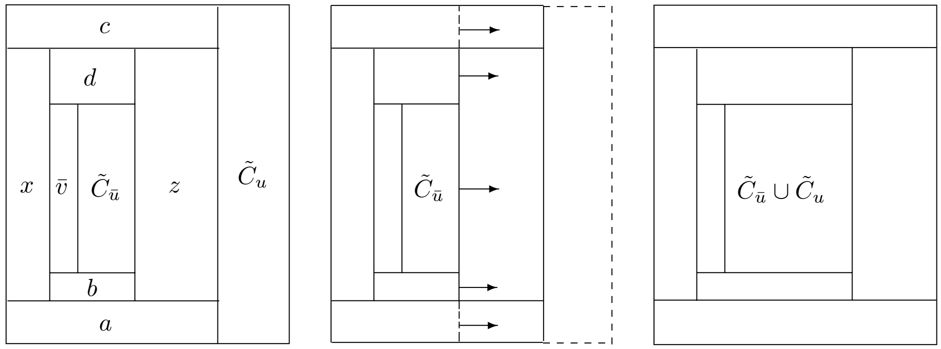


Figure 4. Decreasing the height of the guillotine tree.

of the quality of the packing produced is essentially similar. ∎

4. Algorithm for a fixed number of stages

In this section, we give an algorithm that finds a $k+2$ stage packing that uses $(1+\varepsilon)OPT_k + O(1)$ bins, where OPT_k is the optimum k stage solution. We begin by giving an overview of the algorithm.

Given an input instance R, we first classify rectangles into B, V, H and S as previously (but using different thresholds). Let OPT_k denote the (fractional) optimum k-stage solution restricted to items in B, H and V which is allowed to pack items in H and V fractionally. As always, we will view the packing in each bin as a guillotine tree of depth k with a root on level 0 corresponding to a bin. As is standard in designing approximation schemes, we will apply a series of transformations which simplify the structure of OPT_k without increasing the number of bins used significantly. We then show how to obtain essentially the best "structured" packing in polynomial time.

The first set of transformations uses (standard) ideas from Section 3.3. We show that widths of rectangles in H and heights of rectangles in V can be rounded so that belong to set of constant cardinality. This procedure is identical to that in Section 3.3. Next, if a node in the guillotine tree has a lot of children then most of these children and the subtrees rooted at them can be combined into a single node (supervertex). This results in a tree in which each node has a constant number of children. The procedure is similar to that in Section 3.4.4.

Our second step is to show that there is a set Y of size polynomial in n and that only depends on the input instance and that the following property holds: Given any guillotine packing of the instance, it can be transformed into another guillotine packing where the heights and widths of each rectangular region (corresponding to nodes in guillotine trees) belong to the set Y. This is described in Section 4.3.

In our final transformation we show that given any constant stage guillotine packing, we can transform it to another one in which the heights and widths of rectangular regions lie in a set of constant cardinality, without increasing the number of bins significantly. This transformation crucially uses the fact that the packing is guillotine and that there are a constant number of stages. This is described in Section 4.4. An important point however is that these constant number of heights and widths depend on the particular packing. Hence, our algorithm does not know these heights and widths used by the optimum guillotine packing, when this transformation is applied to it. However we will be able to get around this problem, by using the result described in the paragraph about the set Y.

Based on these observations, our algorithm is the following: From the polynomially many candidates for heights and widths in the set Y, we guess a constant number of these. Call this set X. Clearly, there are polynomially choices for X. For each such choice of X, we then consider all possible valid guillotine trees of constant depth and degree where each node corresponds to a rectangular region whose dimensions are consistent with X. Since each guillotine tree has a constant number of nodes, the set of all possible guillotine trees is bounded by a constant ($|X|$ raised to the number of nodes in a tree). Call this set of possible trees $\mathcal{T}$. For each tree $t \in \mathcal{T}$, our algorithm guesses the number of trees n_t that are used by the optimum solution. As the optimum solution uses at most n bins (one for each rectangle), we have at most polynomially many choices for the tuples $(n_1, n_2, \ldots, n_{\mathcal{T}})$. Finally, we show how to obtain an almost optimum packing, given the right guess of trees used by the optimum packing. We now give the details.

4.1. Classification of Items

Let $\varepsilon > 0$ be our precision parameter and let us assume that $1/\varepsilon$ is an integer. We define a sequence of numbers ε_i, $i = 0, \ldots, 1/\varepsilon$ iteratively as $\varepsilon_0 = 1$, and $\varepsilon_{i+1} = \frac{\varepsilon}{f(\varepsilon_i)}$ where $f(x) = 1/x^k$. We again classify all rectangles in R according to their size as before but with respect to the numbers ε_i, $i = 0, \ldots, 1/\varepsilon$. Define $U_i = \{r \in R | \varepsilon_i < w_r \leq \varepsilon_{i-1} \text{ or } \varepsilon_i < h_r \leq \varepsilon_{i-1}\}$, for $i = 1, \ldots, 1/\varepsilon$. Since there are $1/\varepsilon$ sets U_i and every rectangle appears in at most two of them there is at least one index $p = 1, \ldots, 1/\varepsilon$ such that the total area of rectangles in U_p is at most $2\varepsilon \sum_{r \in R} w_r h_r$.

We call all rectangles in U_p *medium* and pack them separately by NFDH using $O(\varepsilon)OPT + O(1)$ bins. The remaining rectangles are classified into *big* (B), *vertical* (V), *horizontal* (H) and *small* (S) as follows:

$$
\begin{aligned}
B &= \{r \in R | w_r > \varepsilon_{p-1} \quad \text{and} \quad h_r > \varepsilon_{p-1}\}, \\
H &= \{r \in R | w_r > \varepsilon_{p-1} \quad \text{and} \quad h_r \leq \varepsilon_p\}, \\
V &= \{r \in R | w_r \leq \varepsilon_p \quad \text{and} \quad h_r > \varepsilon_{p-1}\}, \\
S &= \{r \in R | w_r \leq \varepsilon_p \quad \text{and} \quad h_r \leq \varepsilon_p\}.
\end{aligned}
$$

4.2. Basic Transformations

Consider the optimum fractional k-stage packing of the instance restricted to B, V and H. As usual, we view the packing in a bin as a guillotine tree of depth k, where each leaf corresponds to a rectangular region that contains exactly one rectangle or a fractional piece of it. Wlog, the cuts can be assumed to alternate at every level (i.e., for any ℓ, if level ℓ nodes are obtained by applying horizontal cuts, then the nodes at level $\ell + 1$ are obtained by applying vertical cuts).

First, as we have a fractional packing, using the argument in Section 3.3, we can assume that there are constant number of distinct heights for rectangles in V and distinct widths for rectangles in H.

Second, we apply the transformation of Section 3.4.4. Consider a guillotine tree T and some vertex u of T. Suppose that the children of u are obtained by applying horizontal guillotine cuts. As all children of u have equal width, we combine the rectangular regions $R_1, \ldots, R_g$ that only contain items from H into one big rectangular region R'. We define a new dummy vertex for R' and delete all $R_1, \ldots, R_g$ and their subtrees. This means that we will not be interested in the structure of guillotine subdivisions inside R'. Observe that all other (non-dummy) children of u contain at least one item either from V or B and therefore have height at least ε_{p-1}, and hence u has at most $1/\varepsilon_{p-1}$ children after this procedure.

4.3. Bounding the possible number of candidate sizes for rectangular regions

We begin by showing that we can round the height of a packing in a dummy vertex containing rectangles in H (or width of a packing in dummy vertex containing rectangles in V) can be rounded to an integral multiple of ε_p. We do it as follows: define a dummy horizontal cut such that the distance between the cut and the top edge of R' is less than or equal to ε_p and the distance between the bottom edge of R' and this cut is an integral multiple of ε_p. Throw away all rectangles that have been packed in the region between the top edge of R' and the new dummy cut. Note that we throw away rectangles with cumulative area at most $2\varepsilon_p$. As the total number of nodes in the modified guillotine tree is at most $(1/\varepsilon_{p-1})^k$, the cumulative area thrown away per tree is at most $2(1/\varepsilon_{p-1})^k \varepsilon_p \leq 2\varepsilon$. Hence the total area thrown away in the instance is at most $2\varepsilon \cdot OPT$ which can be repacked using NFDH in $O(\varepsilon \cdot OPT)$ additional separate bins. Let C denote the set of all widths and heights of rectangles in B, the set of all widths of rectangles in H and the set of all heights of rectangles in V. Clearly, C is a set of cardinality at most $2n$ and each element in C has size at least ε_{p-1}. Let $\tilde{C}$ denote the set of all possible linear combinations of entries in $C \cup \{\varepsilon_p\}$ that sum up to at most 1. Note that $|\tilde{C}| \leq (2n)^{1/\varepsilon_p}$. Observe that $\tilde{C}$ only depends on the input instance and not on some particular packing. For a dummy node, applying the repacking procedure in Figure 3 we can assume that the dimensions of the packing in that node lie in $\tilde{C}$. Our next step, the description and proof of which we omit due to lack of space, is to give a bottom-up procedure which when applied to guillotine trees, guarantees the following:

Lemma 3 *After applying the procedure, each guillotine tree is transformed into another guillotine tree of the same depth and at most twice the number of nodes, such that for each node in the tree that contains at least one rectangle, its height and width lie in $\tilde{C}$.*

4.4. Almost optimum solution with O(1) different rectangular regions

Given any k stage guillotine packing $\mathcal{P}$, we can partition the guillotine trees in $\mathcal{P}$ into two sets depending whether the first stage guillotine cuts are horizontal or vertical. We will describe the rounding procedure for trees for which the first stage cuts are horizontal (the procedure for trees with vertical first stage cuts is analogous). After applying the rounding procedure there are only $O(1)$ distinct heights and widths for rectangular regions corresponding to nodes in the trees and the number of additional bins required does not increase significantly.

Our rounding procedure will consist of k phases. In the first phase, we consider all the rectangular regions in the packing produced by the first stage guillotine horizontal cuts. All these rectangular regions have the same width equal to one. Consider these rectangular regions in the decreasing height order, and let $\{1, \ldots, n_1\}$ be the set of indices such that rectangular region heights are $1 \geq h_1 \geq h_2 \geq \ldots \geq h_{n_1}$. Partition these into $1/(\varepsilon\varepsilon_p^2)$ subsets $G_1, \ldots, G_{1/\varepsilon\varepsilon_p^2}$, such that G_1 contains $\lceil \varepsilon\varepsilon_p^2 n_1 \rceil$ rectangular regions with largest heights, G_2 contains the next $\lceil \varepsilon\varepsilon_p^2 n_1 \rceil$ highest rectangular regions and so on. Round the height of every rectangular region up to the largest height in its group. Observe that we do not change the packing of items inside of a rectangular region, but we just round up its height. We remove the rectangular regions in G_1 from the packing and pack them separately in new bins by NFDH. Since each rectangular region has area at least ε_p^2, we have that $|\mathcal{P}|$, the number of bins used by $\mathcal{P}$, is at least $\varepsilon_p^2 n_1$ and hence the number of additional new bins needed for G_1 is at most $|G_1| = \varepsilon\varepsilon_p^2 n_1 + 1 \leq \varepsilon \cdot |\mathcal{P}| + 1$. For $i = 2, \ldots, 1/(\varepsilon\varepsilon_p^2)$, we now place the rounded rectangular regions from G_i in the previous positions occupied by G_{i-1}. Observe that this gives a feasible packing of these rectangular regions and we now have a packing that uses $(1+\varepsilon) \cdot |\mathcal{P}|$ bins and has only a constant number of different heights for the first level shelves.

Suppose we already applied $q-1$ rounding phases to our packing, for $2 \leq q \leq k$. Let t_q denote the total number of different types of rectangular regions at level $q-1$ after the rounding. We then consider each of these t_q width types (or height types) and apply the procedure applied for in the first stage to each of these types. Clearly, $t_{q+1} \leq t_q \cdot 1/\varepsilon\varepsilon_p^2$ and as $t_2 = 1/\varepsilon\varepsilon_p^2$ it follows that $t_{q+1} \leq (1/\varepsilon\varepsilon_p^2)^q$. We now show that the number of bins used in the new instance after stage $q-1$ rounding phases is at most $(1+(q-1)\varepsilon) \cdot |\mathcal{P}| + F_{q-1}(\varepsilon)$. Here $F_{q-1}(\varepsilon)$ is a function that only depends on ε. Observe that this is satisfied for $q = 2$. Since each rectangular region has area at least ε_p^2, we have that $|\mathcal{P}| \geq \varepsilon_p^2 n_q$ where n_q is the number of rounded rectangular regions on stage q. The total number of rectangular regions moved to the new bins is at most $\varepsilon\varepsilon_p^2 n_q + t_q$ and since each rectangular region can clearly be packed in one bin, the total number of additional bins added is at most $\varepsilon\varepsilon_p^2 n_q + t_q = \varepsilon \cdot \varepsilon \cdot |\mathcal{P}| + t_q$. Thus we have that,

Theorem 2 *There exists an almost optimum fractional packing where there are at most $1/(\varepsilon\varepsilon_p^2)^k$ different types of rectangular regions corresponding to vertices in the guillotine trees.*

4.5. The algorithm

Theorem 2 implies that there is a near optimal packing such that there exists a collection of guillotine trees with the property that the rectangular regions corresponding to the nodes of these trees have a constant number of types. Moreover, by Lemma 3 we have that the possible heights and widths of any rectangular region lie in the set $\tilde{C}$ which is polynomially bounded in n. We will use these facts to obtain an almost optimum solution.

Our algorithm is the following: We guess the set Γ whose size is at most $1/(\varepsilon\varepsilon_p^2)^k$ of the possible types for rectangular regions from the set $\tilde{C} \times \tilde{C}$. For a particular guess of Γ, we consider all possibly modified guillotine trees of depth k which have at most $1/\varepsilon_{p-1}^k$ leaves and every vertex of which corresponds to a rectangular region in Γ. For each leaf of the tree we also mark it B, V or H and we denote these marked trees as tree configurations. Let $\mathcal{T}$ denote the set of all valid tree configurations. There are at most $|3\Gamma|^{1/\varepsilon_{p-1}^k}$ (and hence constant) number of such tree configurations in $\mathcal{T}$.

For each tree configuration $t \in \mathcal{T}$ we guess n_t, the number of trees t used in the optimum solution. Since the optimum solution uses at most n bins (each rectangle can be packed in a separate bin), thus the number of choices is at most the number of solutions to $\sum_{t \in \mathcal{T}} n_t \leq n$ which is at most $(n+|\mathcal{T}|)^{|\mathcal{T}|}$ such choices.

It remains to show how to pack items in B,V,H and finally S in the guessed tree configurations. To pack rectangles in B, we form a bipartite graph with rectangles in B on the left and rectangular regions labelled with B on the right. We then find a perfect matching in this graph. Note that such a matching exists since we have guessed the set of tree configurations used in the optimum solution.

To pack items from V, H and S, we again use the procedure of [14] as in Section 3.5. Here, we give only a brief sketch due to space constraints. As the leaves are already marked either B, V or H, we simply consider all leaves marked V as different strips and write a configuration linear program as in [14] to pack the rectangles in V in these leaves. We apply an analogous procedure for rectangles in H. To bound the number of additional bins required by this procedure, we simply note that the number of leaves in each guillotine tree is a constant, and our choice of the parameters ε_i

allows to bound the area wasted in each leaf such that the total area wasted in a guillotine tree is at most ε. We apply an analogous procedure for rectangles in H.

Finally, the items from S are packed in empty areas by using NFDH. Again the standard argument and our choice of parameters ε_i gives us that if we need additional bins to pack items in S, then all bins except possibly the last have a wasted area of at most ε, which implies the asymptotic approximation scheme.

Finally, we note that total number of stages used by our approximate solution is $k+2$ since if we have a dummy node on level k and items from S in it, it will take two more levels to represent NFDH shelf packing of those items in the guillotine tree.

5. Conclusions

The best known approximation algorithm for the 2BP problem [3] with a guarantee of $T_\infty = 1.691\ldots$, finds a 2-stage packing, and shows that it is at most $1.691\ldots$ times worse than the optimum (general) 2BP solution. Given our result, a natural approach to obtain a better approximation ratio for 2BP would be to show a structural result that the best guillotine packing is never worse than $\alpha < T_\infty$ times the optimum general packing. The worst case instance that we know of has a gap of $4/3$ between the best guillotine packing and optimum general packing.

Using essentially the same (but more involved) techniques in combination with techniques from [4] we can reduce number of levels used by our APTAS from $k+2$ to k. Another variation of our approach yields an APTAS for the guillotine bin packing with orthogonal rotations.

6. Acknowledgment

We would like to thank Jon Lee for suggesting the title of the paper.

References

[1] B. S. Baker, D. J. Brown and H. P. Kasteff. 1981. A 5/4 algorithm for two-dimensional packing. *Journal of Algorithms* 2: 348–368.

[2] N. Bansal, J. Correa, C. Kenyon and M. Sviridenko. Bin Packing in Multiple Dimensions: Inapproximability Results and Approximation Schemes. *Mathematics of Operations Research*, to appear (preliminary versions appeared in two separate *SODA 2004* papers by Bansal and Sviridenko and by Correa and Kenyon).

[3] A. Caprara. 2002. Packing two-dimensional bins in harmony. In *Proceeding of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, 490–499.

[4] A. Caprara, A. Lodi, and M. Monaci. 2005. Fast Approximation Schemes for the Two-Stage, Two-Dimensional Bin Packing Problem. *Mathematics of Operations Research* 30:150–172.

[5] F. R. K. Chung, M. R. Garey, and D. S. Johnson. 1982. On packing two-dimensional bins. *SIAM Journal on Algebraic and Discrete Methods* 3:66–76.

[6] E. G. Coffman, M. R. Garey, and D. S. Johnson. 1996. Approximation algorithms for bin packing: a survey. In *D. Hochbaum, editor, Approximation algorithms for NP-hard problems*, 46–93. PWS Publishing, Boston.

[7] E. G. Coffman, M. R. Garey, D. S. Johnson and R. E. Tarjan. 1980. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing* 9:808–826.

[8] J. Csirik and G.J. Woeginger. 1997. Shelf algorithms for on-line strip packing. *Information Processing Letters* 63:171–175.

[9] J.M.V. De Carvalho and A.J.G. Rodrigues. 1995. A LP-based approach to a two-stage cutting stock problem. *European Journal of Operational Research* 84:580–589.

[10] W. Fernandez de la Vega and G. Lueker. 1981. Bin packing can be solved within $1+\varepsilon$ in linear time. *Combinatorica* 1:349–355.

[11] P. C. Gilmore and R. E. Gomory. 1961. A linear programming approach to the cutting stock problem. *Operations Research* 9:849–859.

[12] P. C. Gilmore and R. E. Gomory. 1965. Multistage cutting problems of two and more dimensions. *Operations Research* 13:94–119.

[13] N. Karmarkar and R. M. Karp. 1982. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proceeding of the 23rd IEEE Symposium on Foundations of Computer Science (FOCS)*, 312–320.

[14] C. Kenyon and E. Rémila. 2000. A Near-Optimal Solution to a Two-Dimensional Cutting Stock Problem. *Mathematics of Operations Research* 25: 645–656.

[15] M.L. Mc Hale and R.P. Shah. 1999. Cutting the Guillotine Down to Size. *PC AI magazine* 13.

[16] J. Puchinger, G.R. Raidl and G. Koller. 2004. Solving a Real-World Glass Cutting Problem. In *Evolutionary Computation in Combinatorial Optimization, 4th European Conference*, Lecture Notes in Computer Science 3004, pp. 165–176.

[17] W. Schneider. 1988. Trim-loss minimization in a crepe-rubber mill; optimal solution versus heuristic in the 2 (3)-dimensional case. *European Journal of Operational Research* 34:273–281.

[18] F. Vanderbeck. 2001. A nested decomposition approach to a 3-stage 2-dimensional cutting stock problem. *Management Science* 47:864–879.