Improved Approximation Algorithm for Two-Dimensional Bin Packing

Nikhil Bansal *
Department of Mathematics and Computer Science
Eindhoven University of Technology, Eindhoven, Netherlands
Email: n.bansal@tue.nl

Arindam Khan †
School of Computer Science
Georgia Institute of Technology, Atlanta, GA, USA
Email: akhan67@gatech.edu

Abstract

We study the two-dimensional bin packing problem with and without rotations. Here we are given a set of two-dimensional rectangular items I and the goal is to pack these into a minimum number of unit square bins. We consider the *orthogonal packing* case where the edges of the items must be aligned parallel to the edges of the bin. Our main result is a 1.405-approximation for two-dimensional bin packing with and without rotation, which improves upon a recent 1.5 approximation due to Jansen and Prädel. We also show that a wide class of rounding based algorithms cannot improve upon the factor of 1.5.

Keywords: Rectangle Packing, Bin Packing, Scheduling and Resource Allocation Problems, Approximation Algorithms, Combinatorial Optimization.

1 Introduction

Bin packing is one of the most fundamental problems in optimization and has been extensively studied in approximation algorithms starting from the classical work of Garey and Johnson [12]. The problem is also important from a practical standpoint and finds various applications in scheduling and routing. In this paper we consider the two dimensional bin packing problem, defined as follows. We are given a collection of rectangular items specified by their width and height, that must be packed into a minimum number of unit size square bins. We consider the widely studied *orthogonal packing* case, where the items must be placed in the bin such that their sides are parallel to the sides of the bin. Here two variants are usually studied, (i) where the items cannot be rotated, and (ii) they can be rotated by 90 degrees.

Already in the 1-D case, a simple reduction from the *Partition* problem shows that it is NP-hard to determine whether a set of items can be packed in two bins or not, implying that no approximation better than 3/2 is possible. However, this does not rule out the possibility of an $\mathrm{Opt}+1$ guarantee, and hence it is insightful to consider the *asymptotic approximation ratio* (AAR, denoted by R_A^∞). Given a poly-time algorithm A, we define $R_A^\infty = \lim_{n\to\infty} \sup R_A^n$, where $R_A^n = \max\{A(I)/\mathrm{Opt}(I) | \mathrm{Opt}(I) = n\}$ and I ranges over all possible problem instances. A problem is said to admit an *Asymptotic Polynomial Time Approximation Scheme* (APTAS) if for every $\epsilon > 0$, there is a poly-time algorithm with an asymptotic approximation ratio of $(1+\epsilon)$.

Related previous work. In their celebrated work, de la Vega and Lueker [11] gave the first APTAS for the 1-D bin packing problem. This was substantially improved by Karmarkar and Karp [20] who gave a guarantee of $\mathrm{Opt} + O(\log^2 \mathrm{Opt})$. Very recently, this has been improved by Rothvoss [26] to $\mathrm{Opt} + \tilde{O}(\log \mathrm{Opt})$. On the other hand, the possibility of an algorithm with an $\mathrm{Opt}+1$ guarantee is still open.

The 2-D case is substantially different from the 1-D case. Bansal et al. [2] showed that no APTAS

*This research has been supported by the NWO grant 639.022.211

†Part of this research and travel was supported by ACO program, ARC Fellowship and P. Tetali's grant NSF DMS 1101447

is possible unless P=NP. On the positive side, there has also been a long sequence of works giving improved algorithms. Until the mid 90's the best known bound was a 2.125 approximation [9], which was improved by Kenyon and Rémila [22] to a $2+\epsilon$ approximation for any $\epsilon > 0$. An important breakthrough was achieved by Caprara [5], who gave an algorithm that achieves an asymptotic approximation ratio of $T_\infty + \epsilon \approx 1.69103 + \epsilon$. Here T_∞ is the well-known "Harmonic" constant that appears ubiquitously in the context of bin packing. This was later improved by Bansal et al. [3] to $(\ln T_\infty + 1) \approx 1.52$ by combining the algorithm of Caprara [5] with a general approximation method for set-covering problems known as *Round-and-Approx*, that we will also consider in this paper.

Recently Jansen and Prädel [16] improved this guarantee further to give a 1.5-approximation algorithm. Their algorithm is based on exploiting several non-trivial structural properties of how items can be packed in a bin. This is the best algorithm known so far, and holds both for the case with and without rotations. We remark that there is still a huge gap between these upper bounds and known lower bounds. In particular, the best known explicit lower bound on the asymptotic approximation for 2-D BP is currently $1+1/3792$ and $1+1/2196$ for the versions with and without rotations respectively [7].

Our Results. Our main result is an improved algorithm for the 2-D bin packing problem. In particular we show the following.

THEOREM 1.1. *There is a polynomial time algorithm with an asymptotic approximation ratio of* $\ln(1.5) + 1 \approx 1.405$ *for 2-D bin packing. This holds both for the version with and without rotations.*

The main idea behind theorem 1.1 is to show that the round and approx framework introduced by [3] (we describe this in section 2) can be applied to the result of Jansen and Prädel [16]. Roughly speaking, this framework states that given a packing problem, if (i) the configuration LP for the problem (with the original item sizes) can be solved up to error $1+\epsilon$ for any $\epsilon > 0$, and (ii) there is a ρ approximation for the problem that is *subset-oblivious*; then one can obtain a $(1 + \ln \rho)$ asymptotic approximation for the problem.

In [3], it was shown that the APTAS for 1-D BP due to [11] and the 2-D BP algorithm of [5] are subset-oblivious. However, the notion of subset-obliviousness as defined in [3] is based on various properties of dual-weighting functions, making it somewhat tedious to apply and also limited in scope (e.g. it is unclear to us how to apply this method directly to the algorithm of [16]).

In this paper we give a more general argument to apply the R&A framework directly to a wide class of algorithms[1], and without any reference to dual-weighting functions. In particular, we show that any algorithm based on rounding the (large) items into $O(1)$ types is subset-oblivious. The main observation is that any ρ-approximation based on rounding the item sizes can be related to another configuration LP (on rounded item sizes) whose solution is no worse than ρ times the optimum solution. As the item sizes are rounded, there are only $O(1)$ constraints in this LP and it can be easily shown to be subset oblivious.

For the particular case of 2-D BP, we present the algorithm of Jansen and Prädel that directly fits in the above framework. As most algorithms for bin-packing problems are based on rounding into $O(1)$ types, this makes the framework widely applicable. For example, this gives much simpler proofs of all the results in [3].

Finally, we give some results to show the limitations of rounding based algorithms in obtaining better approximation ratios. Rounding of items to $O(1)$ types has been used implicitly [4] or explicitly [11, 20, 5, 16, 21], in almost all bin packing algorithms. There are typically two types of rounding: either the size of an item in some coordinate (such as width or height) is rounded up in an instance-oblivious way (e.g. in Harmonic rounding [23, 5], or rounding sizes to geometric powers [20]), or it is rounded up in a input sensitive way (e.g. in linear grouping [11]). We show the following result for 2-D bin packing.

THEOREM 1.2. *Any rounding based algorithm that rounds at least one side of each large item to some number in a constant size collection values chosen independent of problem instance (let us call such rounding input-agnostic), cannot have an approximation ratio better than* 3/2.

Remark: The algorithm in theorem 1.2 is allowed to determine which dimension to round for each item type, based on the problem instance. The only restriction we require is that identical items must be rounded in the same way.

Organization. The paper is organized as follows. In section 2, we present the preliminaries. In section 3,

[1] This includes all known algorithms that we know of for bin-packing type problems, except the ones based on R&A method.

we describe how the Round and Approx framework can be applied to rounding based algorithms. In section 4, we present the 1.5 approximation algorithm of [16] and show how the round and framework applies to it. Finally, in Section 5 we show our lower bounds for rounding based algorithms.

2 Preliminaries

2.1 Configuration LP The best known approximations for most bin packing type problems are based on strong LP formulations called configuration LPs. Here there is a variable for each possible way of feasibly packing a bin (called a *configuration*). This allows the packing problem to be cast as a set covering problem, where each item in the instance I must be covered by some configuration. Let $\mathcal{C}$ denote the set of all valid configurations for the instance I. The configuration LP is defined as:

(2.1)
$$\min\{\sum_{C\in\mathcal{C}} x_C : \sum_{C\ni i} x_C \geq 1(i\in I), x_C \geq 0,\ (C\in\mathcal{C})\}.$$

As the size of $\mathcal{C}$ can possibly be exponential in the size of I, one typically considers the dual of the LP given by:

(2.2)
$$\max\{\sum_{i\in I} v_i : \sum_{i\in C} v_i \leq 1(C\in\mathcal{C}), v_i \geq 0, (i\in I)\}.$$

The separation problem for the dual is the following knapsack problem. Given set of weights v_i, is there a feasible configuration with total weight of items more than 1. From the well-known connection between separation and optimization [14, 24, 15], solving the dual separation problem to within a $(1+\epsilon)$ accuracy suffices to solve the configuration LP within $1+\epsilon$ accuracy.

Note that the configurations in (2.1) are defined based on the original item sizes (without any rounding). However, for more complex problems (say 3-D BP) one cannot hope to solve such an LP to within $1+\epsilon$ accuracy, as the dual separation problem becomes at least as hard as 2-D BP. In general, given a problem instance I, one can define a configuration LP in multiple ways (say where the configurations are based on rounded sizes of items in I, which might be necessary if the LP with original sizes is intractable).

For the special case of 2-D BP, the separation problem for the dual (2.2) is the 2-D geometric knapsack problem for which the best known result is only a 2-approximation. However, Bansal et al. [1] showed that the configuration LP (2.1) with original sizes can still be solved to within $1+\epsilon$ accuracy (this is a non-trivial result and requires various ideas). The fact that solving the configuration LP does not incur any loss for 2-D BP plays a key role in why the R&A framework can give an important improvement for the problem.

2.2 Next Fit Decreasing Height (NFDH) In our algorithm we will heavily use the Next Fit Decreasing Height(NFDH) procedure introduced by Coffman et al. [8]. NFDH considers items in a non-increasing order of height and greedily packs items in this order into *shelves*, where a *shelf* is a row of items having their bases on a line that is either the base of the bin or the line drawn at the top of the highest item packed in the shelf below. More specifically, items are packed left-justified starting from bottom-left corner of the bin, until the next item does not fit. Then the shelf is closed and the next item is used to define a new shelf whose base touches the tallest(left most) item of the previous shelf. If the shelf does not fit into the bin, the bin is closed and a new bin is opened. The procedure continues till all the items are packed. A key property of NFDH that we need is the following.

LEMMA 2.1. *[8] Let B be a rectangular region with width w and height h. If we pack small rectangles (with both width and height less than ϵ) using NFDH into B, total $w\cdot h-(w+h)\cdot\epsilon$ area can be packed, i.e. the total wasted volume in B is at most $(w+h)\cdot\epsilon$.*

2.3 R&A Framework Now we describe the R&A Framework as described in [3], but adapted for the 2-D BP problem.

1. Solve the LP relaxation of (2.1) using the APTAS in [1]. Let x^* be the (near)-optimal solution of the LP relaxation and let $z^* = \sum_{C\in\mathcal{C}} x^*_C$. Let r be the number of configurations in the support of x^*.

2. Initialize a $|\mathcal{C}|$-dimensional binary vector x^r to be a all-0 vector. For $\lceil(\ln\rho)z^*\rceil$ iterations repeat the following: select a configuration $C'\in\mathcal{C}$ at random with probability $x^*_{C'}/z^*$ and let $x^r_{C'} := 1$.

3. Let S be the remaining set of items not covered by x^r i.e. $i\in S$ if and only if $\sum_{C\ni i} x^r_C = 0$. On set S, apply ρ approximation algorithm $\mathcal{A}$ that rounds the items to $O(1)$ types and then pack. Let x^a be the solution returned by $\mathcal{A}$ for the residual instance S.

4. Return $x = x^r + x^a$

Let $\mathrm{Opt}(S)$ and $\mathcal{A}(S)$ denote the value of the optimal solution and the approximation algorithm used to solve the residual instance, respectively. Since the algorithm uses randomized rounding in step 2, the residual instance S is not known in advance. However, the algorithm should perform "well" independent of S. For this purpose [3] define the notion of *subset-obliviousness* where the quality of approximation algorithm to solve the residual instance is expressed using a small collection of vectors in $\mathbb{R}^{|I|}$.

DEFINITION 1. *An asymptotic ρ-approximation for the set covering problem defined in (1), is called subset-oblivious if, for any fixed $\epsilon > 0$, there exist constants k, Λ, β (possibly dependent on ϵ), such that for every instance I of (1), there exist vectors $v^1, v^2 \cdots v^k \in \mathbb{R}^{|I|}$ that satisfy the following properties:*

1. *$\sum_{i \in C} v_i^j \leq \Lambda$, for each configuration $C \in \mathcal{C}$ and $j = 1, 2, \ldots k$;*
2. *$Opt(I) \geq \sum_{i \in I} v_i^j$ for $j = 1, 2, \ldots k$;*
3. *$\mathcal{A}(S) \leq \rho(\max_{j=1}^{k} \sum_{i \in S} v_i^j) + \epsilon Opt(I) + \beta$, for each $S \subseteq I$.*

Roughly speaking, the vectors are analogues of the sizes of items and are introduced to use the properties of the dual of (1). Property 1 says that the vectors divided by constant Λ must be feasible for (2). Property 2 provides lower bound for $\mathrm{Opt}(I)$ and property 3 guarantees that the $\mathcal{A}(S)$ is not significantly larger than ρ times the lower bound in property 2 associated with S.

The main result about the R&A is the following.

THEOREM 2.1. *(simplified) If a problem has a ρ asymptotic approximation algorithm that is subset oblivious, and the configuration LP with original item sizes can be solved to within $(1+\epsilon)$ accuracy in polynomial time for any $\epsilon > 0$, then the R&A framework gives a $1 + \ln \rho$ asymptotic approximation.*

3 R&A Framework for Rounding Based Algorithms

We describe here a general approach to show that a wide class of algorithms for bin-packing type problems, in particular those based on rounding the item sizes to $O(1)$ types is subset-oblivious. While such algorithms are hard to define formally, we state their general approach below which subsumes all the known algorithms that we are aware of.

General form of a rounding based algorithm. A typical rounding based algorithm for a d-dimensional problem has the following form. Given some accuracy parameter $\epsilon > 0$, one first defines two functions $f(\epsilon)$ and $g(\epsilon)$ (that only depend on ϵ) with $g(\epsilon) \ll f(\epsilon)$. Call an item *big* if all its coordinates are at least $f(\epsilon)$, and *small* if all its coordinates are at most $g(\epsilon)$. Call an item *medium* if at least one coordinate lies in the range $(g(\epsilon), f(\epsilon))$.

A standard argument [25, 6] shows that the functions g and f can be chosen such that their ratio is as large as desired, while ensuring that the volume of medium items is at most ϵ times $\mathrm{Vol}(I)$, the total volume of items in the input instance I. These items can be ignored as they can be packed in $O(\epsilon) \cdot \mathrm{Opt}$ separate bins using NFDH. Now, all items have each coordinate either small (in $[0, g(\epsilon)]$) or big (in $[f(\epsilon), 1]$). Call an item *skewed* if it is neither big or small (i.e. some coordinates are less than $g(\epsilon)$ and some more than $f(\epsilon)$). Skewed items can be classified into at most $2^d - 2$ types based on which subset of coordinates is large and d is the total number of coordinates.

Now, the algorithm rounds the large dimensions of big and skewed items to $O(1)$ values (possibly in a very complex way, including guessing of sizes), and only focuses on their packing. The small items are ignored and filled later using NFDH in the empty spaces in the packing of big and skewed items. The large separation between g and f ensures that this incurs negligible loss in volume. Finally, one argues that in any packing almost all skewed items are placed in large regions called containers, where each container satisfies the following: (i) has all items of the same type, (ii) has large size in each dimension and (iii) the items are packed within a container with a negligible loss of volume. Thus these containers can be viewed as big items. Then one defines some algorithm $\mathcal{A}$ that finds a good packing of these rounded big items and containers.

It is easily checked that the algorithm of [16], as stated in section 4, falls directly in this framework. We remark that the rounding of sizes in their algorithm is non-trivial and actually depends on which the bin patterns are used in the optimum solution (that the algorithm will guess).

Relating the algorithm to the configuration LP with rounded item sizes. Fix some packing problem, and suppose $\mathcal{A}$ is a rounding based ρ-approximation algorithm for it. Then, $\mathcal{A}(I) \leq \rho \cdot \mathrm{Opt}(I)$ for any instance I of the problem. Let $\tilde{I}$ de-

note instance obtained from I by rounding the large dimensions according to the rounding performed by $\mathcal{A}$. Clearly, $\mathcal{A}(I) \geq \mathrm{Opt}(\tilde{I})$ and hence

$$\mathrm{Opt}(\tilde{I}) \leq \rho \cdot \mathrm{Opt}(I). \tag{3.3}$$

Now, consider the configuration LP defined on the instance $\tilde{I}$, where the configurations correspond to feasible packing of the big items and the containers in $\tilde{I}$. As there are only a constant number of item types, this LP has only a constant number t of non-trivial constraints, one for each item type.

For concreteness, let us consider the case of 2D-BP. The items are classified into big and small. There are two types of skewed items: long (with height $\geq f(\epsilon)$ and width $\leq g(\epsilon)$) and wide (with width $\geq f(\epsilon)$, height $\leq g(\epsilon)$). Upon rounding, the big items are rounded to $O(1)$ types and long (and wide) items are assigned to $O(1)$ groups of different heights (or widths). For $i = 1, \ldots, p_1$, let B_i denote the group of big items rounded to type i and $c^r_{B_j}$ be the number items of type B_j in the r'th configuration. Long items are rounded to p_2 heights and assigned to groups $L_1, \cdots L_{p_2}$. Similarly wide items are rounded to p_3 widths and assigned to groups $W_1, \cdots W_{p_3}$. Let $w(L_k)$ denote the total width of items in L_k, and $h(W_\ell)$ denote the total height of items in W_ℓ. Similarly, let $c^r_{L_k}$ (resp. $c^r_{W_\ell}$) denote the total width of items in L_k (resp. total height of items in W_ℓ) in the configuration r.

Consider the following configuration LP: $\mathrm{LP}(\tilde{I})$

$$\begin{aligned}
\text{Min} \quad & \sum_r x_r & & \\
\text{s.t.} \quad & \sum_r c^r_{B_j} x_r & \geq & |B_j| \quad \forall j \in [p_1] \\
& \sum_r c^r_{L_k} x_r & \geq & w(L_k) \quad \forall k \in [p_2] \\
& \sum_r c^r_{W_\ell} x_r & \geq & h(W_\ell) \quad \forall \ell \in [p_3] \\
& x_r & \geq & 0 \quad (r = 0, 1 \ldots m)
\end{aligned}$$

Let $t = p_1 + p_2 + p_3$. As the LP has only t constraints,

$$\mathcal{A}(\tilde{I}) \leq \mathrm{LP}^*(\tilde{I}) + t. \tag{3.4}$$

Furthermore, let us assume that the right hand side for each constraint in the LP above is either 0, or is at least $\Omega((1/\epsilon^2)\log t)$. If this is not the case, we simply remove these items from the configurations and pack them in new bins using NFDH. This requires at most $O(t \cdot (1/\epsilon^2)\log t) = O_\epsilon(1)$ additional bins. This property will be useful later.

3.1 R&A for rounding based algorithms We can now show the following.

THEOREM 3.1. *If there is a ρ approximation algorithm $\mathcal{A}$ that rounds the large coordinate of items to $O(1)$ types before packing (these sizes could depend on the instance I), then the R&A method gives a $(1 + \ln \rho)$ asymptotic approximation bin packing for I.*

Proof. First, we consider the configuration LP in step 1 of the R&A framework and apply the randomized rounding step to it. The probability that an item $i \in I$ is not covered by $x_{C'}$ in some iteration, is $1 - \sum_{C \ni i} x^*_C / z^*$. Let S be the set of residual items not covered by any of the bins selected in $(\ln \rho) z^*$ iterations. Thus the probability that i is not covered in any of the $(\ln \rho) z^*$ iterations is at most:

$$\begin{aligned}
\mathbb{P}(i \in S) &= (1 - \sum_{C \ni i} x^*_C / z^*)^{\lceil (\ln \rho) z^* \rceil} \\
&\leq (1 - \sum_{C \ni i} x^*_C / z^*)^{(\ln \rho) z^*} \leq e^{(-\ln \rho)} = \frac{1}{\rho}.
\end{aligned} \tag{3.5}$$

where the last inequality follows as $\sum_{C \ni i} x^*_C \geq 1$ for all $i \in I$ and $(1 - x^{-1})^{\alpha x} \leq e^{-\alpha}$ for $x > 0$.

Let $\mathrm{Opt}(I)$ be the number of bins used in the optimal packing of I. Now in step 2 at most $\lceil (\ln \rho) z^* \rceil \leq 1 + (\ln \rho) \cdot \mathrm{Opt}$ bins were used. Let S denote the set of items that are still unpacked. It remains to bound the number of bins used for packing S using $\mathcal{A}$.

To this end, consider the rounding that $\mathcal{A}$ would apply to the items when given instance I, and consider the instance obtained by applying this rounding to items in S. Let us denote this instance as $\tilde{I} \cap S$. Now consider the following configuration LP for $\tilde{I} \cap S$: $\mathrm{LP}(\tilde{I} \cap S)$

$$\begin{aligned}
\text{Min} \quad & \sum_r x_r & & \\
\text{s.t.} \quad & \sum_r c^r_{B_j} x_r & \geq & |B_j \cap S| \quad \forall j \in [p_1] \\
& \sum_r c^r_{L_k} x_r & \geq & w(L_k \cap S) \quad \forall k \in [p_2] \\
& \sum_r c^r_{W_l} x_r & \geq & h(W_\ell \cap S) \quad \forall \ell \in [p_3] \\
& x_r & \geq & 0 \quad (r = 0, 1 \ldots m)
\end{aligned}$$

Now by (3.5), $\mathbb{E}[|B_j \cap S|] \leq |B_j|/\rho$, and similarly $\mathbb{E}[w(L_k \cap S)] \leq w(L_k)/\rho$ and $\mathbb{E}[h(W_\ell \cap S)] \leq h(W_\ell)/\rho$. Moreover, as each $|B_j|, w(L_k)$ and $h(W_\ell)$ are at least $\Omega((1/\epsilon^2)\log t)$, by standard Chernoff bounds

it follows that the probability that $|B_j \cap S| \geq (1+\epsilon)\mathbb{E}[|B_j \cap S|]$ is at most $\exp(-\epsilon^2|B_j|/\rho) = \exp(-\Omega(\log t)/\rho) = 1/\text{poly}(t)$. Similarly, this holds for all other constraints. Taking a union bound over the t constraints, it follows that with high probability, the right hand side for each constraint in $\text{LP}(\tilde{I} \cap S)$ is at most $(1+\epsilon)/\rho$ times the right hand side of the corresponding constraint in $\text{LP}(\tilde{I})$. This gives us that,

$$\begin{aligned}\mathcal{A}(\tilde{I} \cap S) &\leq \text{LP}^*(\tilde{I} \cap S) + t \\ &\leq \frac{(1+\epsilon)}{\rho}\text{LP}(\tilde{I}) + O(1) \\ &\leq \frac{(1+\epsilon)}{\rho}\text{Opt}(\tilde{I}) + O(1) \\ &\leq (1+\epsilon)\text{Opt}(I) + O(1).\end{aligned}$$

Here, the first step follows by (3.4) and the last step follows by (3.3). This gives the desired $1 + \ln \rho$ asymptotic approximation.

The above algorithm can be derandomized using standard techniques as in [3].

4 1.5-approximation algorithm that rounds items to $O(1)$ types

In this section we present Jansen-Prädel algorithm that rounds the items into $O(1)$ types of sizes before packing them into bins. Most of the technical details are moved to the Appendix.

4.1 Technique The algorithm works in two stages. In the first stage, the items in the input instance are rounded to $O(1)$ types of rectangles. By guessing structures of the rounded items, we guess the rounded values and how many items are rounded to each such value. In the second stage rounded rectangles are packed into bins. The algorithm uses the following structural theorem.

THEOREM 4.1. *[16] For any value ϵ_c, with $\frac{1}{\epsilon_c}$ being a multiple of 24, and for any solution that fits into m bins, the widths and the heights of the rectangles can be rounded up so that they fit into $(3/2 + 5\epsilon_c)m + 37$ bins, while the packing of each of the bins satisfies either Property 1.1 (The width of each rectangle in bin $\mathcal{B}_i$ of width at least ϵ_c is a multiple of $\frac{\epsilon_c^2}{2}$) or Property 1.2 (The height of each rectangle in bin $\mathcal{B}_i$ of height at least ϵ_c is a multiple of $\frac{\epsilon_c^2}{2}$).*

Using the above structural theorem they show that given any optimal packing they can remove all items intersected with a thin strip in the bin and round one side of all remaining items to some multiple of $\epsilon_c^2/2$. Then they pack the cut items separately to get a packing in at most $(3/2) \cdot \text{Opt}$ bins that satisfy either property 1.1 or property 1.2. After rounding one side of the rectangle, the other side is rounded using techniques similar to those used by [22]. In this version of the algorithm after items are rounded to $O(1)$ types, we can find the optimal packing of these rounded items by brute-force. The algorithm is actually guessing the structure of optimal packing i.e. rounded values for each item, to use the structural theorem to get a feasible packing in $\leq (\frac{3}{2} + \epsilon)\text{Opt} + O(1)$ bins. In the Appendix, we explain the algorithm step by step and show that the algorithm always rounds the items to $O(1)$ number of types. The main structure of their algorithm is described below:

Input: A set of items $I := \{r_1, r_2, \cdots, r_n\}$ where $r_j \in (0,1] \times (0,1]$ for all $j \in [n]$ and set of bins of size 1×1.
Output An orthogonal packing of I without rotations.

Algorithm:
1. **Guess Opt:** Guess Opt by trying all values between 1 and n.
2. *For* each guessed values of Opt *do*

(a) **Classification of rectangles:** Compute δ using methods similar to [18] to classify rectangles and pack medium rectangles using Next Fit Decreasing Height (NFDH).

(b) **Guessing structures:** Enumerate suitably over all structures (sizes to which items are rounded to and for each size the number of items that are rounded to that size) of the set of big, long, wide rectangles and the set of wide and long containers.

(c) *For* each guess *do* **Packing:**
- Assign big rectangles by solving flow network with the algorithm of Dinic[10];
- Do greedy Assignment of long and wide rectangles into $O(1)$ groups;
- Pack $O(1)$ number of groups of long and wide rectangles into containers by brute force;
- Pack the small rectangles using Next Fit Decreasing Height;
- Pack $O(1)$ types of containers and $O(1)$ types of big rectangles into bins using brute force;

3. *Return* a feasible packing;

5 Lower bound for rounding based algorithms

In this section, we describe some limitations of rounding based algorithms.

Theorem 1.2. (restated) Any rounding algorithm that rounds at least one side of each large item to some fixed constant independent of the problem instance (let us call such rounding input-agnostic), cannot have an approximation ratio better than 3/2.

Proof. Consider an input-agnostic algorithm $\mathcal{A}$ that rounds at least one side of each large item to one of the values $c_1, c_2 \ldots, c_z$, that are chosen independent of the input instance. Let i and j be such that $c_i < 0.5 \leq c_{i+1}$ and $c_{j-1} \leq 0.5 < c_j$. Let $f = \min\{0.5 - c_i, c_j - 0.5\}$. Here we assume the algorithm rounds identical items with the same height and width to same types.

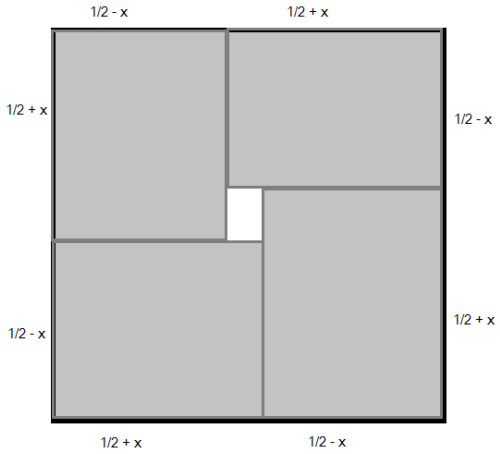

Figure 1: Lower bound example for rounding based algorithms

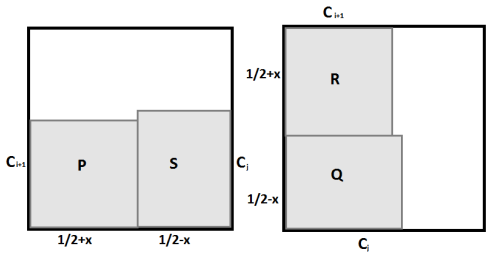

Figure 2: The case when $C_{i+1} > 1/2$

Now consider an optimum packing using $m = 2k$ bins where each bin is packed as in figure 1, for some fixed $x \in (0, f)$. Under the rounding, an item $(1/2 + x) \times (1/2 - x)$ is rounded to either $(1/2+x) \times (c_{i+1})$ (let us call such items of type P) or to $(c_j) \times (1/2 - x)$ (let us call such items of type Q). Similarly, each item $(1/2 - x) \times (1/2 + x)$ is rounded to either $(c_{i+1}) \times (1/2 + x)$ (call these of type R) or to $(1/2 - x) \times (c_j)$ (call these of type S).

Let us first consider the easy case when $c_{i+1} > 1/2$. It is easily checked that in this case, any bin can contain at most 2 rounded items: (i) either a P-item and a S-item or (ii) a Q-item and a R-item. See, for example figure 2. This implies that a 2-approximation is the best one can hope for if 1/2 is not included among the $c_1, c_2 \ldots, c_z$.

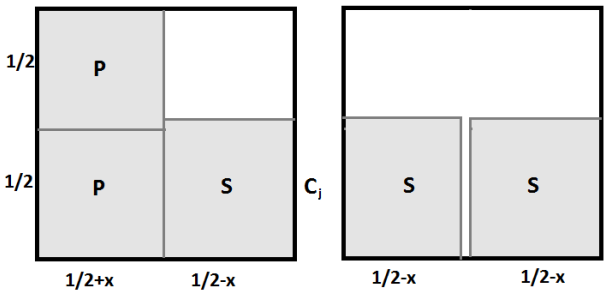

Figure 3: Configuration $\{P, P, S\}$

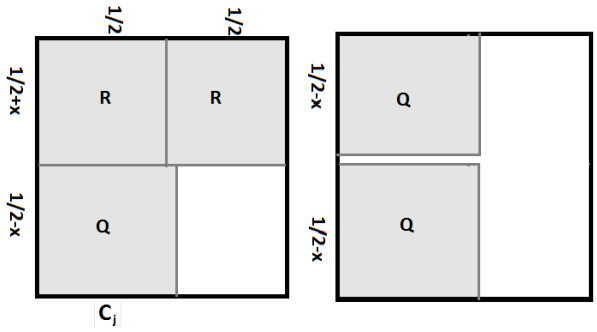

Figure 4: Configurations $\{R, R, Q\}$

We now consider the case when $c_{i+1} = 1/2$. We claim that the possible bin configurations are a) [{ P,P,S } and { S,S }], which happens when the items are rounded to types P and S. See figure 3. Or, b) [{ R,R,Q } and { Q,Q }], which happens when items are rounded to types R and Q. See figure 4. Furthermore, the remaining two cases can be ignored. That is, when items are rounded to type P and R or when items are rounded to type Q and S, as in these cases at most two items can be packed in a bin.

So, let us consider case (a). The proof for case (b) is analogous. Let X_1 and X_2 denote the number of configurations of type { P,P,S } and { S,S } respectively. Then we get the following configuration LP:

$$\begin{aligned} \text{Min} \quad & X_1 + X_2 & & \\ \text{s.t.} \quad & 2X_1 & \geq & 4k \\ & X_1 + 2X_2 & \geq & 4k \\ & X_1, X_2 & \geq & 0 \end{aligned}$$

The dual is:

$$\begin{aligned} \text{Max} \quad & 4k(v_1 + v_2) & & \\ \text{s.t.} \quad & 2v_1 + v_2 & \leq & 1 \\ & 2v_2 & \leq & 1 \\ & v_1, v_2 & \geq & 0 \end{aligned}$$

A feasible dual solution is $v_1 = 0.25, v_2 = 0.5$. This gives dual optimal as $\geq 3k$. Thus the number of bins needed is $\geq 3k = 3m/2$.

This in particular implies that to beat 3/2 one would need a rounding that is not input-agnostic, or which rounds identical items with the same height and width to different types, sometimes rounded by width and sometimes by height.

We also note that 4/3 is the lower bound for any rounding algorithm that rounds items to $O(1)$ types. This seems to be a folklore observation, but we state it here for completeness and give a proof in the appendix.

THEOREM 5.1. *Any algorithm that rounds items to $O(1)$ types cannot achieve better than 4/3 approximation.*

6 Final Remarks

The approach for the R&A framework described here applies directly to wide variety of algorithms and gives much simpler proofs for previously considered problems (e.g. vectorBP, 1D BP) [3]. As rounding large coordinates to $O(1)$ number of types is by far the most widely used technique in bin-packing type problems, we expect wider applicability of this method.

Moreover, improving our guarantee for 2-D BP will require an algorithm that is not input-agnostic. In particular, this implies that it should have the property that it can round two identical items (i.e. with identical height and width) differently. One such candidate is the guillotine packing approach [4]. It has been conjectured that this approach can give an approximation ratio of 4/3. One way to show this would be to prove a structural result bounding the gap between guillotine and non-guillotine packings. At present the best known upper bound on this gap is $T_\infty \approx 1.69$ [6].

Acknowledgements We thank Prasad Tetali for helpful discussions.

References

[1] Nikhil Bansal, Alberto Caprara, Klaus Jansen, Lars Prädel and Maxim Sviridenko, *A Structural Lemma in 2-Dimensional Packing, and Its Implications on Approximability*, In ISAAC(2009), pp. 77-86. 3

[2] Nikhil Bansal, José R. Correa, Claire Kenyon and Maxim Sviridenko, *Bin Packing in Multiple Dimensions: Inapproximability Results and Approximation Schemes*, Mathematics of Operations Research(2006), pp. 31(1):31–49. 1

[3] Nikhil Bansal, Alberto Caprara and Maxim Sviridenko, *A New Approximation Method for Set Covering Problems, with Applications to Multidimensional Bin Packing*, SIAM Journal of Computing(2009), pp. 39(4):1256-1278. 2, 3, 4, 6, 8

[4] Nikhil Bansal, Andrea Lodi and Maxim Sviridenko, *A Tale of Two Dimensional Bin Packing*, In FOCS(2005), pp. 657–666. 2, 8

[5] Alberto Caprara, *Packing 2-Dimensional Bins in Harmony*, In FOCS(2002), pp. 490–499. 2

[6] Alberto Caprara, Andrea Lodi and Michele Monaci, *Fast Approximation Schemes for Two-Stage, Two-Dimensional Bin Packing*, Mathematics of Operations Research(2005), pp. 30(1):150-172. 4, 8

[7] Miroslav Chlebík and Janka Chlebíková, *Inapproximability Results for Orthogonal Rectangle Packing Problems with Rotations*, In CIAC(2006), pp. 199-210. 2

[8] Edward G. Coffman Jr., Michael R. Garey, David S. Johnson and Robert E. Tarjan, *Performance Bounds for Level-Oriented Two-Dimensional Packing Algorithms*, SIAM Journal of Computing(1980), pp. 9(4):808-826. 3

[9] Fan Chung and Michael R. Garey and David S. Johnson, *On packing two-dimensional bins*, SIAM Journal of Algebraic Discrete Methods(1982), pp. 3:66-76. 2

[10] Yefim Dinic, *An algorithm for solution of a problem of maximum flow in a network with power estimation*, Soviet Mathematics Doklady(1970), pp. 11(5):12771280. 6, 11

[11] Wenceslas Fernandez de la Vega and George S. Lueker, *Bin packing can be solved within 1+epsilon in linear time*, Combinatorica(1981), pp. 1(4):349-355. 1, 2, 10

[12] Michael R. Garey and David S. Johnson, *Computers and Interactibility: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Company, CA(1979). 1

[13] Michael R. Garey and David S. Johnson, *Approximation Algorithms for Bin packing Problems: A Survey*, Analysis and Design of Algorithms in Combinatorial Optimization, Springer-Verlag, Berlin(1981).

[14] Michael D. Grigoriadis, Leonid G. Khachiyan, Lorant Porkolab and Jorge Villavicencio, *Approximate Max-Min Resource Sharing for Structured Concave Optimization*, SIAM Journal on Optimization(2001), pp. 11(4):1081-1091. 3

[15] Martin Grötschel, László Lovász and Alexander Schrijver, *Geometric algorithm and combinatorial optimization*, Algorithms and Combinatorics: Study and Research Texts, 2. Springer-Verlag, Berlin(1988). 3

[16] Klaus Jansen and Lars Prädel, *New Approximability Results for Two-Dimensional Bin Packing*, In SODA(2013), pp. 919-936. 2, 3, 4, 6, 9, 11

[17] Klaus Jansen, Lars Prädel and Ulrich M. Schwarz, *Two for One: Tight Approximation of 2D Bin Packing*, In WADS(2009), pp. 399-410.

[18] Klaus Jansen and Roberto Solis-Oba, *Rectangle packing with one-dimensional resource augmentation*, Discrete Optimization, pp. 6(3):310–323. 6, 9, 12

[19] Ravi Kannan, *Minkowskis convex body theorem and integer programming*, Mathematics of Operations Research(1987), pp. 12(3):415440. 12

[20] Narendra Karmarkar and Richard M. Karp, *An Efficient Approximation Scheme for the One-Dimensional Bin-Packing Problem*, In FOCS(1982), pp. 312-320. 1, 2

[21] David R. Karger and Krzysztof Onak, *Polynomial approximation schemes for smoothed and random instances of multidimensional packing problems*, In SODA(2007), pp. 1207-1216. 2

[22] Claire Kenyon and Eric Rémila, *A Near-Optimal Solution to a Two-Dimensional Cutting Stock Problem*, Mathematics of Operations Research(2000), pp. 25(4):645-656. 2, 6, 10, 12

[23] Chan C. Lee and Der-Tsai Lee, *A Simple On-Line Bin-Packing Algorithm*, Journal of ACM(1985), pp. 32:(3):562-572. 2

[24] Serge A. Plotkin and David B. Shmoys and Eva Tardos, *Fast approximate algorithm for fractional packing and covering problems*, Mathematics of Operations Research(1995), pp. 20:257-301. 3

[25] Lars Prädel, *Approximation Algorithms for Geometric Packing Problems*, PhD Thesis, Kiel, Christian-Albrechts-Universitat(2012). 4

[26] Thomas Rothvoß, *Approximating Bin Packing within $O(\log Opt * \log\log Opt)$ bins*, In FOCS(2013). 1

A Appendix

A.1 Details of Algorithm in Section 4: In this section we describe the algorithm in [16], to fit in our framework.

A.1.1 Binary Search for Opt: Using binary search between the number of rectangles (upper bound) and total area of the rectangles (lower bound), the algorithm finds the minimum m such that there exists a feasible solution with $(3/2+\epsilon)\cdot m + O(1)$ bins. For each guess of Opt, we first guess the rounding in the following way and then we pack the rounded items into the bins.

A.1.2 Classification of rectangles: A value δ ($\leq \epsilon' = \epsilon/48$) is selected similar to [18], such that $\frac{1}{\delta}$ is a multiple of 24 and rectangles with at least of one of the side lengths between δ and δ^4 have a small area ($\leq \epsilon' \cdot \text{Opt}$). Now we classify the rectangles into five types:

- Big: both width and height is at least δ.
- Wide: width is at least δ, height is smaller than δ^4.
- Long: height is at least δ, width is smaller than δ^4.
- Small: both width and height is less than δ^4.
- Medium: either width or height is in $[\delta^4, \delta)$. As medium rectangles are of total size $\leq \epsilon' \cdot \text{Opt}$, they can be packed using NFDH into at most additional $O(\epsilon' \cdot \text{Opt})$ bins. Choose $\epsilon_c = \delta$.

A.2 Rounding and Guessing structures: First we will show that given any optimal packing, we can get a packing in at most $(\frac{3}{2}+\epsilon)\text{Opt}+O(1)$ bins where all items are rounded to $O(1)$ types. Then we will guess the structure of optimal packing to assign items to its rounded type.

Assuming we are given the optimal packing, we can get rounding of one side by using theorem 4.1. Now we will find the rounding of the other side. Let Opt $= m$ and out of these m bins m_1 bins $\mathcal{B}_1, \mathcal{B}_2 \cdots \mathcal{B}_{m_1}$ are of type 1 (that satisfy property 1.1, i.e. the width and the x-coordinate of each rectangle in $\mathcal{B}_i$ of width at least ϵ_c is a multiple of $\frac{\epsilon_c^2}{2}$) and remaining $m_2 (= m - m_1)$ bins $\mathcal{B}_{m_1+1}, \mathcal{B}_{m_1+2} \cdots \mathcal{B}_m$ are of type 2 (that satisfy Property 1.2, i.e. the height and the y-coordinate of each rectangle in $\mathcal{B}_i$ of height at least ϵ_c is a multiple of $\frac{\epsilon_c^2}{2}$). Thus the widths of big and wide rectangles in bins of type 1 and the

heights of big and long rectangles in bins of type 2 are rounded to some multiples of $\delta^2/2$. Let B_i^w and W_i^w are the set of big and wide rectangles respectively that are packed in type 1 bin and widths are rounded to $i \cdot \delta^2/2$ for $i \in \{2/\delta, 2/\delta + 1, \cdots, 2/\delta^2\}$. Similarly, let B_i^h and L_i^h are the set of big and long rectangles respectively that are packed in type 2 bin and heights are rounded to $i \cdot \delta^2/2$. Let L^w and W^h be the set of long rectangles in type 1 bin and set of wide rectangles in type 2 bins respectively. Set of small and medium rectangles are denoted by M and S respectively.

A.2.1 Rounding of big, long and wide rectangles: The rounded widths of rectangles in B_i^w and W_i^w and rounded heights of rectangles in B_i^h and L_i^h are known. In this step we find the rounding of heights of rectangles in B_i^w and L^w and rounding of widths of rectangles in B_i^h and W^h using linear grouping techniques similar to Kenyon-Rémila [22] and introduced by Fernandez de la Vega and Lueker [11] .

For any set B_i^w, we sort the items $r_i^1, r_i^2 \cdots r_i^{|B_i^w|}$ according to non-increasing height, i.e. $h(r_i^e) \geq h(r_i^f)$ for $e \leq f$. Now define at most $\frac{1}{\delta^2}$ subsets $B_{i,j}^w$, each of which contains $\lceil \delta^2 \cdot |B_i^w| \rceil$ rectangles except possibly for the last subset. For any two rectangles $r_A \in B_{i,j_1}^w$ and $r_B \in B_{i,j_2}^w$ and $j_2 \geq j_1$, $h(r_A) \geq h(r_B)$. We round height of all rectangles in each set $B_{i,j}^w$ to the height of tallest rectangle in the subset (we call it to be the round rectangle of the subset). Set apart the set $B_{i,1}^w$ and for other rectangles in $B_{i,j}^w$ place them on the position of rectangles of $B_{i,j-1}^w$. This is possible as all subsets(except possibly the last) have same cardinality and all rectangles have same width.

Similarly, sort long rectangles in L^w according to non-increasing height. We divide the set L^w into at most at most $\frac{1}{\delta^2}$ subsets $L_1^w \cdots L_{1/\delta^2}^w$ such that every subset has total width $\delta^2 \cdot w(L^w)$. If needed items are sliced vertically to create subsets. For any two rectangles $r_A \in L_{j_1}^w$ and $r_B \in L_{j_2}^w$ and $j_2 \geq j_1$, $h(r_A) \geq h(r_B)$. We round the height of each rectangle to the height of the tallest rectangle in it. Apart from L_1^w, rectangles of L_j^w are packed on position of rectangles of L_{j-1}^w.

The rectangles in $L_1^w, B_{2/\delta,1}^w, \cdots B_{2/\delta^2,1}^w$ are packed separately into additional bins using NFDH. Note that width of all rectangles in $L^w, B_{2/\delta}^w, \cdots B_{2/\delta^2}^w$ is at most $1/\delta.m_1$ as each rectangles has height at least δ. So, $w(L_1^w) + w(B_{2/\delta,1}^w), \cdots w(B_{2/\delta^2,1}^w) \leq \delta^2.w(L^w) + w(B_{2/\delta}^w), \cdots w(B_{2/\delta^2}^w) \leq \delta^2 \cdot 1/\delta \cdot m_1 = \delta \cdot m_1$. Thus the total area of the rectangles in $L_1^w \cup B_{2/\delta,1}^w \cup \cdots \cup B_{2/\delta^2,1}^w$ is $O(\delta \cdot m_1)$ and thus can be packed in additional $O(\delta \cdot m_1)$ bins using NFDH.

Widths of rectangles in B_i^h, W^h are rounded in a similar manner.

A.2.2 Rounding of containers: We have not rounded width of long rectangles and heights of wide rectangles. Now we construct rectangular containers for the wide and long rectangles for that purpose. We only show the rounding of containers for type 1 bins. Rounding containers for type 2 bins can be done analogously. Let C_L^w be the set of containers for long rectangles and C_W^w be the set of containers for wide rectangles in type 1 bins. Define $2/\delta^2$ vertical slots of width $\delta^2/2$ in each type-1 bin $\mathcal{B}_i$. A long container is part of a slot that contains at least one long rectangle, and the container is bounded at the top and bottom by a wide or big rectangle or the boundary(ceiling or floor) of the bin. There can be at most $(1/\delta - 1)$ long containers in a slot. Thus there are at most $O(\delta^3)$ long containers per bin.

Next, construct wide containers by extending upper and lower edges of big rectangles and long containers in both directions till they hit another big rectangles, long container or boundary (left or right side of bin). Wide and small rectangles are horizontally cut by these lines. As there are $O(\delta^3)$ big rectangles and long containers, there are $O(\delta^3)$ wide containers in $\mathcal{B}_i$. This way any packing in optimal bin is transformed into a packing of big rectangles and long and wide containers. There is no empty space left, hence all small rectangles are fractionally in the long and wide containers.

Now we do the rounding of containers. Heights of all containers in C_W^w are rounded down to nearest multiple of δ^4 cutting the uppermost wide and short rectangles. There are $O(1/\delta^3) \cdot m_1$ wide containers and small rectangles have height less than δ^4. Thus the cut wide and short rectangles are packed using NFDH in additional $O(\delta \cdot m_1)$ bins. For long containers we remove the short rectangles and push all long rectangles vertically down till they touch top of another rectangle or boundary. Then we round down the heights to either nearest multiple of δ^4 or combination of rounded heights of the long rectangles. Note that these heights are rounded down to although large but still $O(1)$ number of types. Total area loss for each container is $O(\delta^4 \cdot \delta^2/2)$ and number of long containers is $O(1/\delta^3)$. So in the reduced container we pack the small items till we can and the remaining small rectangles are packed into additional

$O(\delta^3 \cdot m_1)$ bins using NFDH.

Similarly long containers can be constructed for the additional bins that are used to pack items of L_1^w. These bins will have at most $(2\delta \cdot p_1 + 1) \cdot 2/\delta^2$ long containers of width $\delta^2/2$ and height 1. Note that there are $O(1/\delta^2)^{1/\delta}$ possible height of containers, which can be reduced to $O(1/\delta^2)$ heights using linear grouping and loosing only a small constant.

Thus at the end of the rounding of containers, the containers have the following properties:
2.1. There are at most $O(1/\delta^3) \cdot m_1$ wide containers in C_W^w with width of multiple of δ^2 and height of a multiple of δ^4.
2.2. There are at most $O(1/\delta^3) \cdot m_2$ long containers in C_L^h with height of multiple of δ^2 and width of a multiple of δ^4.
2.3. There are at most $O(1/\delta^3) \cdot m_2$ wide containers in C_W^h with $O(1/\delta^2)$ different widths(either a multiple of δ^4 or a combination of rounded width of wide rectangles in W^h) and height $\delta^2/2$.
2.4. There are at most $O(1/\delta^3) \cdot m_2$ long containers in C_L^w with $O(1/\delta^2)$ different heights(either a multiple of δ^4 or a combination of rounded height of long rectangles in L^w) and width $\delta^2/2$.

At the end of the rounding step we get the following theorem.

THEOREM A.1. *[16] Given an optimal packing I into m bins, it is possible to round the widths and heights of the long, wide and big rectangle to $O(1)$ types such that it fits in at most $(3/2 + O(f_1(\delta)))m + O(f_2(\delta))$ bins for some functions f_1 and f_2 and these bins satisfy either property 1.1 or property 1.2. Furthermore the heights of long and big rectangles in L^w and B^w, widths of wide and big rectangles in W^h and B^h are rounded up to $O(1/\delta^2)$ values. The wide and long rectangles are sliced horizontally and vertically respectively and packed into containers satisfying properties 2.1-2.4 and small rectangles are packed fractionally into the wide and long containers. Medium rectangles are packed separately in $O(\delta)$ bins.*

A.2.3 Transformation of rectangles: Now we guess the structure of the optimal packing for the assignment of rectangles to the rounded rectangles.

First we have to determine whether width or height of a big rectangle is rounded to a multiple of $\delta^2/2$. We guess the cardinality of sets B_i^w and B_i^h for $i \in \{2/\delta, 2/\delta + 1, \cdots 2/\delta^2\}$. This can be done by choosing less than $2 \cdot (2/\delta^2)$ values out of n and note that this is polynomial in n. For each such guess we also guess $2 \cdot (2/\delta^2) \cdot (1/\delta^2)$ round rectangles out of n rectangles. These values give us the structure of subsets as discussed in the rounding of big rectangles. Now to find the assignment of big rectangles to these subsets, we create a directed flow network $G = (V, E)$. First we create source(s) and target node(t). For each rectangle $r \in I$, we create a node and add an edge from s to r with capacity one. Next we create nodes for all subsets $B_{i,j}^w$ and $B_{i,j}^h$ and add an edge from r to $B_{i,j}^y$ of capacity one if r might belong to $B_{i,j}^y$ where $y \in \{w, h\}$. Next add edges between nodes corresponding to subsets and the target node of infinite capacity. Now apply Dinic's algorithm [10] or any other flow algorithm to find if there is a $s-t$ flow of value same as the number of big rectangles. If there exists such a flow, we get a valid assignment of big rectangles into subsets. On the other hand, if there is no such flow then continue to other guesses.

Now we need to transform the wide and long rectangles. First we need to decide whether a wide rectangle belong to type 1 or type 2 bins. The case for long rectangle is analogous. Note that in the linear grouping of wide rectangles in W^h, $1/\delta^2$ subsets were created. The total height of all rectangles in W^h is bounded by $n \cdot \delta^4$. So we approximately guess the total height of W^h by choosing some $t \in \{1, 2, \cdots n\}$ so that $t \cdot \delta^4 \leq h(W^h) \leq (t+1) \cdot \delta^4$. As all subsets $W_1^h, W_2^h, \cdots W_{1/\delta^2}^h$ have same height, each subset will have height $h(W^h) \cdot \delta^2$. We also guess the height of rectangles to which all rectangles in each subset are rounded to. This can be done by choosing $1/\delta^2$ rectangles out of n rectangles. Thus we can approximately guess the structure of the rectangles in W^h. Remaining wide rectangles are in W^w i.e. their width are rounded to next multiple of $\delta^2/2$. We guess approximately the total height of the rectangles in $W_{2/\delta}^w, \cdots W_{2/\delta^2}^w$ by choosing $(2/\delta^2 - 2/\delta + 1)$ integral values t_k such that $t_k \cdot \delta^4 \leq h(W_j^w) < (t_k + 1) \cdot \delta^4$. Thus we can guess structure of all subsets of wide rectangles and we need to assign the wide rectangles into these subsets. For the assignment we sort the wide rectangles in non-increasing order. We assign wide rectangles greedily into all sets $W_k^w \in W^w$ starting from W_{2/δ^2}^w till total height exceeds $(t_k+1) \cdot \delta^4$ for each set W_k^w. The remaining rectangles are similarly greedily assigned into sets $W_1^h, \cdots W_{1/\delta^2}^h$. It is easy to show that for the right guesses of t_k values, all rectangles will have a valid assignment. Afterwards we remove the shortest rectangles in each subset to reduce the height to at most $t_k \cdot \delta^4$. It can be shown that the total height of these removed wide rectangles is $O(\delta^2)$ and thus can be packed into $O(1)$ additional bins.

A.2.4 Construction of containers: Here we describe the construction of long and wide containers that are placed in type-1 bins. The construction of long and wide containers that are placed in type-2 bins, is analogous.

Each wide containers in C_W^w has height of a multiple of δ^4 and width of multiple of $\delta^2/2$. Hence we can guess $n_{i,j}^w$, number of wide containers that has width $i\delta^2/2$ and height $j \cdot \delta^4$ by choosing $1/\delta^4 \cdot 2/\delta^2$ values out of n.

Similarly long containers in C_L^w have same width and $O(1/\delta^2)$ types of heights (either a combination of rounded heights of long rectangles or a multiple of δ^4) that we can guess.

A.2.5 Packing long and wide rectangles into containers There are four cases: packing of long rectangles into long containers in type 1 bins, packing of long rectangles into long containers in type 2 bins, packing of wide rectangles into wide containers in type 1 bin and packing of wide rectangles into wide containers in type 2 bins. Here let us only consider the wide containers in type 1 bins, other cases can be handled similarly. As there are $O(\delta^3)$ types of wide containers and $O(\delta^2)$ types of wide rectangles, we try out all possible way wide rectangles can be put into wide containers by brute force. The running time can be further improved if we use linear programs as in Kenyon-Rémilla [22]. Then we add back the small rectangles using NFDH in the empty regions of the $O(\delta^3)$ types of containers till we can.

A.2.6 Complete packing Now we have big rectangles and containers of $O(1)$ type, thus there are $O(1)$ number of possible configurations of packing of big rectangles and containers into bins. We try out all configurations by brute force to find the optimal packing of big rectangles and containers. Then we add back the small rectangles using NFDH in the empty regions of the bin till we can and the remaining small rectangles are packed into additional bins.

A.3 Analysis In the rounding step, separate packing of rectangles in $L_1^w, B_{2/\delta,1}^w, \cdots B_{2/\delta^2,1}^w, W_1^h, B_{2/\delta,1}^h, \cdots B_{2/\delta^2,1}^h$ need at most $O(\delta\text{Opt})$ additional bins. In rounding containers the cut wide, long and small rectangles are packed in additional $O(\delta\text{Opt})$ bins. Packing of medium and remaining small rectangles take $O(\delta\text{Opt})$ bins. Removed wide rectangles in the step of transformation of wide rectangles require $O(1)$ extra bins. So using the structural theorem total $(3/2 + O(\delta))\text{Opt} + O(\delta)$ bins are sufficient.

The running time of the steps are given as follows. The binary search requires $O(\log n)$ time. Computing δ in a method similar to [18] takes $O(n/\epsilon)$ time. For the structure of the set of big rectangles, we guess $O(1/\delta^2)$ values out of n to guess the cardinality of the sets and for such guess, $O(1/\delta^4)$ round rectangles are guessed. Similarly, we get the structure of wide and long rectangles, we guess $O(1/\delta^3)$ values out of n. Structure of long and wide containers require guessing $O(1/\delta^6)$ values out of n and guessing $O(1/\delta^2)$ values out of $O(1/\delta^4 + (1/\delta^2)^{1/\delta})$ respectively. Solving the flow network takes $O(n^3)$ time. Assignment of wide and long rectangles into groups will take $O(n \log n)$ time. The running time for packing containers and big rectangles using the brute force method is a large, however, constant in triple exponential in δ. It can be reduced using integer programs of Kannan et al. [19]. Packing medium and small rectangles using NFDH require $O(n \log n/\delta^3)$ time. Totally the running time is bounded by $O(n^{h_1(1/\epsilon)}.h_2(1/\epsilon))$, where h_1, h_2 are polynomial functions. Thus the total running time is polynomial for fixed ϵ.

A.4 Bin packing with rotations Bin packing is rotation is almost similar to the packing without rotation. In this case we only have bins with a packing that satisfy property 1.1. Remaining rounding steps are analogous to the versions with rotations. The step of transformation of rectangles, however, is slightly different when we allow rotations. For big rectangles, in the flow network we connect a big rectangle with all subsets that can contain the rectangle before and after rotating by 90°. On the other hand, for transformation of wide and long rectangles, we approximately guess $w(L^w)$ and the heights of the sets $W_{2/\delta}^w, \cdots W_{2/\delta^2}^w$ and the height of the cut rectangles in L^w. Now we can rotate all long rectangles to have only wide rectangles and greedily assign them to the wide rectangles in $W_{2/\delta}^w, \cdots W_{2/\delta^2}^w$. The remaining wide rectangles are rotated back and assigned to the sets $L_1^w \cdots L_{1/\delta^2}^w$. The analysis is also similar, however, gives slightly better constants in the approximation ratio.

A.5 Proof of theorem 5.1

THEOREM A.2. *If an algorithm rounds items to $O(1)$ types, then it can not achieve better than 4/3 approximation.*

Proof. Consider the packing in figure 1. Assume there is an optimal packing of $m = 3k$ bins where each bin is having similar packing as in figure 1 for m different values of $x \in (0.001, 0.01]$. Note that the sum of the height and width is exactly 1 for each rectangle. If we use rounding to $O(1)$ items, then for all but $O(1)$ items i, $w(i) + h(i)$ exceed 1. Without loss of generality, we assume each item touches boundary. Otherwise for these set of items, we can extend sides vertically and horizontally so that it touches boundary or another item. As the total sum of the side lengths of a bin is 4 and each item has intersection with boundary of length > 1 , we can only pack 3 rounded items in a bin. Thus $4m = 12k$ items can be packed in at least $4k - O(1) = 4/3m - O(1)$ bins. This example packing is particularly interesting as it also achieves the best known lower bound for guillotine packing i.e. 4/3.