RC22900 (W0309-095) September 17, 2003
Computer Science

IBM Research Report

New Approximability and Inapproximability Results for 2-Dimensional Bin Packing

Nikhil Bansal
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Maxim Sviridenko
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

New Approximability and Inapproximability results for 2-dimensional bin packing

Nikhil Bansal[*] Maxim Sviridenko[†]

Abstract

We study the 2-dimensional generalization of the classical Bin Packing problem: Given a collection of rectangles of specified size (width, height), the goal is to pack these into minimum number of square bins of unit size.

A long history of results exists for this problem and its special cases [3, 14, 10, 18, 9, 1, 15]. Currently, the best known approximation algorithm achieves a guarantee of 1.69 in the asymptotic case (i.e. when the optimum uses a large number of bins) [1]. However, an important open question has been whether 2−dimensional bin packing is essentially similar to the 1−dimensional case in that it admits an asymptotic polynomial time approximation scheme (APTAS) [8, 13] or not? We answer the question in the negative and show that the problem is APX hard in the asymptotic case.

On the other hand, we give an asymptotic PTAS for the special case when all the rectangles to be packed are squares (or more generally hypercubes). This improves upon the previous best known guarantee of 1.454 for $d = 2$ [9] and $2 - (2/3)^d$ for $d > 2$ [15], and settles the approximability for this special case.

1 Introduction

In the *2-Dimensional Bin Packing Problem* rectangles of specified size (width, height) have to be packed into larger squares (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. The goal is to find a feasible packing, i.e. a packing where rectangles do not overlap, using smallest number of bins.

Bin packing and its $d-$dimensional variants have been extensively studied since the 60's both in the context of offline approximation algorithms and online algorithms. A detailed survey can be found in [4, 7]. Throughout this paper we only consider offline algorithms, and give only the relevant results.

We assume that every rectangle p has width $1 \geq w_p > 0$ and height $1 \geq h_p > 0$. Clearly the $\mathcal{NP}$-Hardness of 2-dimensional bin-packing follows for that of 1-dimensional bin packing (which is the special case, when all the heights are exactly 1). The standard measure used to analyze the performance of a packing algorithm A is the asymptotic approximation ratio R_A^∞ defined by

$$R_A^n = \max\{A(L)/OPT(L)|OPT(L) = n\}$$

$$R_A^\infty = \lim_{n\to\infty} \sup R_A^n$$

where L ranges over the set of all problem instances and $A(L)$ (resp. $OPT(L)$) denote the number of bins used by A (resp. the optimum algorithm).

For 1-dimensional bin packing, De La Vega and Lueker gave the first asymptotic approximation scheme [8]. Later, this was improved by Karmarkar and Karp to give an algorithm which uses $Opt + O(\log^2 Opt)$ bins [13].

For the 2-dimensional case, the first results were obtained by [3] who gave a 2.125 approximation algorithm. For a long time this was the best known, until a $2 + \epsilon$ (for any $\epsilon > 0$) approximation was obtained (implicitly) by Kenyon and Remila [14]. The recent breakthrough is an elegant 1.691 approximation algorithm due to Caprara [1]. Interestingly, many more results are known for some special cases in which there is a restriction on how the rectangles can be packed in a bin. Two particular cases that are widely studied are *Strip Packing* and *Shelf Packing* (details about these can be found in [1] and the references there in). While clever asymptotic approximations schemes are known for some of these special cases, it was unclear whether the general 2-dimensional bin packing problem admits an approximation scheme.

For the special case of packing squares in square bins (which is also NP-hard by [16]) only algorithms with constant factor approximation ratios were known prior to our work. The first guarantee better than 2.125 was obtained by [10] who gave a 1.988-approximation algorithm. This was later improved to $14/9+\epsilon = 1.55+\epsilon$ by Seiden and Stee [18]. Recently, Caprara gave an

[*]Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: nikhil@cs.cmu.edu.

[†]IBM T.J. Watson Research Center, Yorktown Heights, 10598, Email: sviri@us.ibm.com.

algorithm and showed that it has approximation in the interval $[1.490, 1.507]$, provided a conjecture is true [1]. The best known guarantee prior to our work is due to Epstein and Stee [9] who give an $16/11 + \epsilon = 1.454 + \epsilon$ algorithm for the 2-dimensional case. Finally, for the general $d-$dimensional hypercube packing case, [15] obtained the first algorithm with approximation ratio $2 - (2/3)^d$, for fixed d.

A related problem is that of *Vector Bin packing* described as follows: Given a set of n rational vectors $p_1, \dots, p_n$ from $[0,1]^d$, find a partition of the set into sets $A_1, \dots, A_m$ such that $||\bar{A}_i||_\infty \le 1$ for $1 \le i \le m$, where $\bar{A}_i = \sum_{j \in A_i} p_j$ is the sum of the vectors in A_i. The objective is to minimize m, the size of the partition.

For $d = 1$, the vector bin packing problem is identical to the classical 1-dimensional bin packing, but this is not true for $d > 1$. Chekuri and Khanna [2] showed a relatively simple connection between d dimensional vector bin packing (for arbitrary d) and graph coloring, which implies that vector bin packing is hard to approximate within $O(d^{1-\epsilon})$ for any $\epsilon > 0$. Woeginger [20] showed that the problem is APX hard even for the case of $d = 2$. The best known result for this problem is an $(1 + \epsilon d + O(\ln \epsilon^{-1}))$-approximation for any fixed $\epsilon > 0$ [2]. This in particular implies an $O(\ln(d))$ approximation for constant d.

Another closely related problem is the packing problem where we are given a set of rectangles with nonnegative profits. The goal is to maximize the total weight of rectangles which could be packed into a bigger rectangle (or square by scaling). The most recent paper on this problem is due to Jansen and Zhang [11]. They describe few constant factor approximation algorithms for this problem and provide references on the state of the art for it.

1.1 Our Results

We show the following results:

1. The $2-$dimensional bin packing problem does not admit an asymptotic approximation scheme. This trivially implies the non-existence of an asymptotic PTAS for all $d \ge 2$.

2. We give the first asymptotic approximation scheme for packing squares into square bins. More generally, we give an APTAS for packing d-dimensional hypercubes into bins for any fixed d. Independently of our result, Correa and Kenyon [6] obtained the same result which is published in this proceedings. They also designed a resource augmented PTAS for the general 2-dimensional rectangle packing problem, i.e. the algorithm which packs rectangles into the optimal number of bins of size $(1 + \varepsilon) \times (1 + \varepsilon)$.

1.2 Techniques

Our result for the APX hardness of 2-dimensional bin packing has ideas similar to those used by Woeginger [20] to show the APX hardness of 2 dimensional vector packing. However, our construction is much more involved than in [20] as there is much less structure on how the rectangles can be packed in a bin. For example, given a collection of rectangles (or equivalently vectors) $p_1, \dots, p_n$, it is NP-Hard to decide whether these rectangles can be packed in a single bin in the sense of rectangular bin packing (this follows by a simple reduction from the 3-Partition problem). On the other hand, this problem is trivial for vector bin packing.

For $d > 1$ dimensional square (or hyper-cube) packing, most previous approaches which obtain a constant factor approximation [10, 18, 15, 9] use the classical techniques used for 1-dimensional bin packing. That is, classify the objects into large and small. Find a packing of the large objects using rounding and exhaustive search, and then pack the remaining small objects.

In the case when $d > 1$, the above approach does not directly yield an approximation scheme for the following reason: The gaps left in the bin after packing the large objects can have arbitrary structure, and it is not clear how to pack the small objects in these gaps without wasting a constant fraction of the space.

Our approach is based on a technique due to Sevastianov and Woeginger [19]. We partition the objects into 3 sets: large, medium and small. This gives us a sufficient gap between the sizes of the large objects and the small objects. We pack the medium objects separately, and then show how to pack the large and the small objects together.

2 MAX SNP Hardness of 2 dimensional bin packing

We give a reduction from the Maximum Bounded 3-Dimensional matching problem.

Input: Three sets $X = \{x_1, \dots, x_q\}, Y = \{y_1, \dots, y_q\}$ and $Z = \{z_1, \dots, z_q\}$. A subset $T \subseteq X \times Y \times Z$ such that any element in X, Y, Z occurs in one, two or three triples in T. Note that this implies that $q \le |T| \le 3q$.

Goal: Find a maximum cardinality subset T' of T such that no two triples in T' agree in any coordinate.

Measure: Cardinality of T'.

Kann [12] was first who proved MAX SNP hardness of the MAX-3-DM. Petrank [17] (Theorem 4.4) proved a refined hardness result, he proved that it is NP-hard to distinguish between instances where $|T'| = q$ and instances where $|T'| \le (1-\varepsilon)q$ for some constant $\varepsilon > 0$.

We start with an instance I of MAX-3-DM and we will construct an instance of 2-dimensional bin packing with $5q + 3|T|$ rectangles.

We first define the following integers. Let $r = 32q$.

$$\begin{aligned} x'_i &= ir^3 + i^2 r + 1, \quad \text{for} \quad 1 \le i \le q, \\ y'_i &= jr^6 + j^2 r^4 + 2, \quad \text{for} \quad 1 \le j \le q, \\ z'_i &= kr^9 + k^2 r^7 + 4, \quad \text{for} \quad 1 \le k \le q. \end{aligned}$$

For every triple $t_l = (x_i, y_j, z_k)$ in T, we define an integer $t'_l = r^{10} - x'_i - y'_j - z'_k + 15 = r^{10} - kr^9 - k^2r^7 - jr^6 - j^2r^4 - ir^3 - i^2r + 8$. Let $\delta = 1/500$ and let $c = (r^{10} + 15)/\delta$. Observe that $0 < x'_i, y'_j, z'_k < \delta c/10$ and $t'_l < \delta c$ holds for all i, j, k, l.

We now describe the rectangles in our instance. A rectangle of width w and height h will be denoted by (w, h).

For each element in $x_i \in X$ we define two rectangles $a_{x,i} = (1/4 - 4\delta + x'_i/c, 1/2 + 4\delta - x'_i/c)$ and $a'_{x,i} = (1/4 + 4\delta - x'_i/c, 1/2 - 4\delta + x'_i/c)$

For each element in $y_i \in Y$ we define two rectangles $a_{y,i} = (1/4 - 3\delta + y'_i/c, 1/2 + 3\delta - y'_i/c)$ and $a'_{y,i} = (1/4 + 3\delta - y'_i/c, 1/2 - 3\delta + y'_i/c)$

For each element in $z_i \in Z$ we define two rectangles $a_{z,i} = (1/4 - 2\delta + z'_i/c, 1/2 + 2\delta - z'_i/c)$ and $a'_{z,i} = (1/4 + 2\delta - z'_i/c, 1/2 - 2\delta + z'_i/c)$

Let $A_x = \{a_{x,1}, \ldots, a_{x,q}\}$ and $A'_x = \{a'_{x,1}, \ldots, a'_{x,q}\}$. A_y, A'_y, A_z and A'_z are defined similarly to be the set of rectangles $a_{y,i}$, $a'_{y,i}, a_{z,i}$ and $a'_{z,i}$ respectively. We will use A to denote the collection $A_x \cup A_y \cup A_z$ and $A' = A'_x \cup A'_y \cup A'_z$.

Next for each $t_l \in T$ we define two rectangles b_l and b'_l such that $b_l = (1/4 + 8\delta + t'_l/c, 1/2 + \delta - t'_l/c)$ and $b'_l = (1/4 - 8\delta - t'_l/c, 1/2 - \delta + t'_l/c)$.

Let $B = \{b_1, \ldots, b_{|T|}\}$ and $B' = \{b'_1, \ldots, b'_{|T|}\}$.

Finally we define D to be a collection of $|T| - q$ dummy rectangles d_i such that $d_i = (3/4 - 10\delta, 1)$.

We say that two rectangles a and a' are buddies iff $\{a, a'\}$ is $\{a_{x,i}, a'_{x,i}\}$ or $\{a_{y,j}, a'_{y,j}\}$ or $\{a_{z,k}, a'_{z,k}\}$ for $1 \le i, j, k \le k$ or $\{a, a'\} = \{b_l, b'_l\}$ for some $1 \le l \le |T|$. From the values chosen for the sizes of the rectangles it is easy to see that,

Observation 1. *For each rectangle $a \in A \cup A'$, $w(a) + h(a) = 3/4$. For any two buddies $b \in B$ and $b' \in B'$, $w(b) + h(b) + w(b') + h(b') = 3/2$.*

Observation 2. *For any two rectangles a, a' in $A \cup A' \cup B \cup B'$, $h(a) + h(a') = 1$ iff a and a' are buddies.*

The following definition is needed only for the next two lemmas. For a rectangle a, we now define $\Delta(a)$ which allows us to relate back the rectangle to the integers x'_i, y'_j, z'_k or b'_l. For a rectangle $a_{x,i} \in A_x$, let $\Delta(a_{x,i}) = x'_i$. Similarly, $\Delta(a_{y,j}) = y'_j$, $\Delta(a_{z,k}) = z'_k$, $\Delta(b_l) = t'_l$ and $\Delta(a'_{x,i}) = -x'_i$,$\Delta(a'_{y,j}) = -y'_j$, $\Delta(a'_{z,k}) = -z'_k$, $\Delta(b'_l) = -t'_l$.

Lemma 2.1. *For any three rectangles $a_1, a_2, a_3 \in A$ and $b \in B$, $w(a_1) + w(a_2) + w(a_3) + w(b) = 1$ iff $\{a_1, a_2, a_3, b\} = \{a_{x,i}, a_{y,j}, a_{z,k}, b_l\}$ such that $t_l = (x_i, y_j, z_k)$ is a tuple in the MAX-3-DM problem.*

Proof. The "if" direction follows directly from definitions. We now proof the "only if" part of the lemma. As $0 < \Delta(a_i) < \delta c/10$, for $1 \le i \le 3$ and $0 < \Delta(b) < \delta c$, then if $w(a_1) + w(a_2) + w(a_3) + w(b) = 1$, it must be that $\sum_{i=1}^{3} \Delta(a_i) + \Delta(b) = \delta c = r^{10} + 15$. Considering the quantity $\sum_{i=1}^{3} \Delta(a_i) + \Delta(b)$ modulo r, it follows that there is exactly one rectangle each from A_x, A_y, A_z and B since $15 = 1 + 2 + 4 + 8$ and this is the only way to represent 15 as a sum of four numbers from the set $\{1, 2, 4, 8\}$. Next, considering the sum $\sum_{i=1}^{3} \Delta(a_i) + \Delta(b)$ again modulo $r^2, r^3, \ldots, r^9$, it follows that $\{a_1, a_2, a_3, b\} = \{a_{x,i}, a_{y,j}, a_{z,k}, b_l\}$ such that $t_l = (x_i, y_j, z_k)$ is a tuple in the MAX-3-DM problem.

Lemma 2.2. *Let $\mathcal{R} = \{a_1, a_2, a_3, a_4\}$ be four rectangles lying in $A \cup A'$, such that no two of these are buddies. Then for any choice of a_i, $1 \le i \le 4$, $\sum_i w(a_i) \ne 1$.*

Proof. Suppose for the sake of contradiction that $\sum_{i=1}^{4} w(a_i) = 1$. As $0 \le |\Delta(a_i)| \le \delta c/10$, it must be that $\sum_{i=1}^{4} \Delta(a_i) = 0$. We first show that there cannot be more then one rectangle from $A_x \cup A'_x$ in the set $\mathcal{R}$ (later we will show that same argument works for $A_y \cup A'_y$ and $A_z \cup A'_z$, thus giving a contradiction). Considering the coefficient of r and r^3 in the sum quantity $\sum_{i=1}^{4} \Delta(a_i)$. These coefficients depend on rectangles in $(A_x \cup A'_x) \cap \mathcal{R}$ only. Let i_1, i_2, i_3, i_4 denote the indices of rectangles from $(A_x \cup A'_x) \cap \mathcal{R}$ (where an index is 0, if fewer than 4 occur). Since no two rectangles are buddies, we cannot have both a_{x,i_k} and a'_{x,i_k} in $\mathcal{R}$. So, for each i_k, $1 \le k \le 4$, we associate a variable $\alpha(i_k)$, where $\alpha(i_k) = 1$ if $a_{x,i_k} \in \mathcal{R}$ and -1 if $a'_{x,i_k} \in \mathcal{R}$.

Since the coefficients of r and r^3 sum to 0, we must have that $\sum_{k=1}^{4} \alpha(i_k) i_k = 0$ and $\sum_{k=1}^{4} \alpha(i_k) i_k^2 = 0$, with the constraints that all positive i_k's are distinct.

We now claim that the only feasible solution to the above system is $i_k = 0$ for all $1 \le k \le 4$. Clearly we need at least 3 of the i_k's to be positive else we cannot have $\sum_k \alpha(i_k) i_k = 0$. Second, note that all $\alpha(i_k)$ can not be -1 or +1.

Under these constraints, when exactly three are non-zero i_k's, we have the equations, $i_1 + i_2 = i_3$ and $i_1^2 + i_2^2 = i_3^2$ under the constraints that all numbers are positive and distinct. But clearly, there can be no solution to this.

In the case when all the i_k's are positive by renaming variables all cases can be reduced to the following two cases:

1. $i_1 + i_2 = i_3 + i_4$ and $i_1^2 + i_2^2 = i_3^2 + i_4^2$,
2. $i_1 + i_2 + i_3 = i_4$ and $i_1^2 + i_2^2 + i_3^2 = i_4^2$.

The last case clearly does not have a solution. For the first case, rewrite the equations as $i_1 - i_3 = i_4 - i_2$ and $i_1^2 - i_3^2 = i_4^2 - i_2^2$, which implies that $i_1 + i_3 = i_4 + i_2$, and hence $i_1 = i_4$ which gives a contradiction. Repeating the argument identically for $A_y \cup A'_y$ and $A_z \cup A_z$ the result follows.

OBSERVATION 3. *The width (resp. height) of any rectangle in the instance is at least* $1/4 - 10\delta$ *(resp. at least* $1/2 - 5\delta$*), and hence the area of any rectangle is at least* $1/8 - 25/4\delta > 1/9$*. Thus no bin can have more than 8 rectangles.*

OBSERVATION 4. *If we consider a feasible bin packing as a packing of some unit square with the left lower corner in the origin and right upper corner in the point* $(1, 1)$ *then any vertical line, i.e. a line parallel to the y-axis, intersects at most one rectangle from* $A \cup B$ *since height of each rectangle in* $A \cup B$ *is strictly greater than 1/2.*

This observation implies that any bin can contain at most 4 rectangles in $A \cup B$, and at most 3 rectangles in B since the width of each rectangle in $A \cup B$ more than 1/5, and the width of each rectangle in B is more than 1/4. Finally, it is easy to see that,

OBSERVATION 5. *If a rectangle* $d_i \in D$ *lies in some bin* S*, then* S *contains at most* 2 *other rectangles and at most one of them is a rectangle from* $A \cup B$*.*

Given a packing of the bins, call a bin *good* if it contains exactly 8 rectangles and moreover it has exactly 4 rectangles from $A \cup B$. The following crucial lemma characterizes the structure of good bins.

LEMMA 2.3. *A bin is good if and only if it contains the rectangles* $a_{x,i}, a_{y,j}, a_{z,k}, b_l$ *and the corresponding rectangles* $a'_{x,i}, a'_{y,j}, a'_{z,k}, b'_l$ *such that* $t_l = (x_i, y_j, z_k)$ *corresponds to a triple in MAX-3-DM instance.*

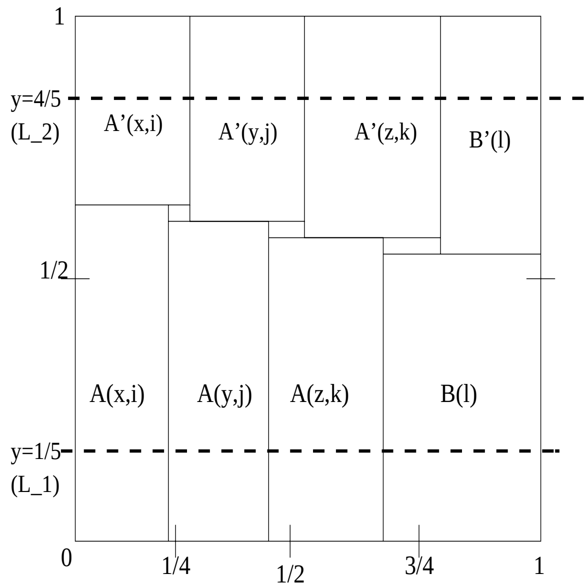

Figure 1: *Packing of the rectangles corresponding to a triple*

Proof. We first show that the rectangles corresponding to a triple can be packed in a bin. Starting from the bottom left corner of the bin and moving towards the right, we pack the rectangles $a_{x,i}$,$a_{y,j}$, $a_{z,k}$ and b_l. Each of these rectangles is placed such that it touches the bottom of the bin. Figure 1 shows the packing. It is easy to verify that these four rectangles can be packed as described, as $w(a_{x,i}) + w(a_{y,j}) + w(a_{z,k}) + w(b_l) = 1 - \delta + x'_i + y'_j + z'_k + t'_l = 1$. Next we observe that each of the rectangles $a'_{x,i}, a'_{y,j}, a'_{z,k}$ and b'_l can be placed in the remaining gaps (as shown in Figure 1). Clearly, $a'_{x,i}$ can be placed on top of $a_{x,i}$ because $h(a_{x,i}) + h(a'_{x,i}) = 1$ and as $h(a_{x,i}) < h(a_{y,j}) < h(a_{z,k}) < h(b_l)$, this allows $a'_{x,i}$ to extend horizontally beyond $a_{x,i}$. Arguing similarly and observing that $w(a'_{x,i}) + w(a'_{y,j}) + w(a'_{z,k}) + w(b'_l) = 1$, it is easy to see that rectangles $a'_{y,j}, a'_{z,k}$ and b'_l also fit.

We now show that any good bin must correspond to a triple. We first give a way for labelling the 8 rectangles in a good bin. Consider the lines $L_1 = \{y = 1/5\}$ and $L_2 = \{y = 4/5\}$. It is easy to see that any in packing of a bin with 8 rectangles, each rectangle must intersect exactly one of L_1 or L_2. This follows as any rectangle has height at most $1/2 + 1/50 < 3/5$ and at least $1/2 - 1/50 > 2/5$. Moreover, as any rectangle has width strictly more than 1/5, it follows that each L_1 and L_2 intersect exactly 4 rectangles. Let $\{a_1, a_2, a_3, a_4\}$ denote the rectangles that intersect L_1 such that a_i is to the left of a_j for $i < j$. Similarly let $\{a_5, a_6, a_7, a_8\}$ denote the rectangles that intersect L_2 in the left to right order. Thus we have that

$$\sum_{i=1}^{4} w(a_i) \leq 1 \tag{2.1}$$

and that

$$\sum_{i=1}^{4} w(a_{i+4}) \leq 1 \tag{2.2}$$

Finally observe that for each $1 \leq i \leq 4$ each rectangle a_i must overlap with a_{i+4} in an $x - coordinate$. Thus we have the constraints that

$$h(a_i) + h(a_{i+4}) \le 1 \qquad \text{for } 1 \le i \le 4 \tag{2.3}$$

We consider three cases depending on the number of rectangles in B that lie in the bin.

1. *At least 2 rectangles in B lie in the bin:* Since each bin in B has width at least $1/4 + 8\delta$, two such bins use at least $1/2 + 16\delta$. Thus the width left for bins from A is at most $1/2 - 16\delta$ (we cannot put rectangles from A on top of the rectangles from B), and hence at most 1 bin from A can fit. Thus the bin can have at most 3 rectangles from $A \cup B$ and hence cannot be good. Similarly, if 3 rectangles from B lie in the bin, there cannot be any rectangle from A.

2. *No rectangle in B lies in the bin:* We claim the bin cannot have more than 3 rectangles from A. For the sake of contradiction suppose $r_1, r_2, r_3, r_4 \in A$ lie in the bin. Then no rectangle from B' can lie in the bin, because any rectangle in A has height at least $1/2 + 2\delta$ while the height of any rectangle in B' is at least $1/2 - \delta$ and moreover any rectangle in B' must overlap in some x-coordinate with some r_i for $1 \le i \le 4$. Thus all the 8 rectangles lie in $A \cup A'$.

   Adding Equations 2.1, 2.2 and 2.3 we have that $\sum_{i=1}^{8}(w(a_i) + h(a_i)) \le 6$. Moreover from Observation 1 we have that $\sum_{i=1}^{8}(w(a_i)+h(a_i)) = 6$, thus it must be the case that each of Equations 2.1,2.2 and 2.3 must hold with an equality. By Observation 2, this implies that a_i and a_{i+4} are buddies for each $i = 1, \ldots, 4$. This in particular implies that among the rectangles a_1, a_2, a_3 and a_4 no two are buddies. Therefore, it is impossible that $\sum_{i=1}^{4} w(a_i) = 1$ by Lemma 2.2, and hence we have a contradiction.

3. *Exactly one rectangle $b_l \in B$ lies in the bin:* In the case we will show that if the bin is good, then it must correspond to a triple $t_l = (x_i, y_j, z_k)$.

   Suppose the bin is good, then there are exactly three other rectangles from A. Since any rectangle from B' cannot overlap in an x-coordinate with any rectangle in A, it follows that there can be *at most* one rectangle from B', which must overlap with b_l. Next, we show that there has to be *at least* one rectangle from B'. Suppose there are no rectangles from B', then we claim that the total width of all the rectangles must be strictly more than 2, which is not possible. To see this, since there is no rectangle from B', there must be four rectangles from A'. As the width of any rectangle in A' (resp. B) is strictly more than $1/4 + \delta$ (resp. $1/4 + 8\delta$), this takes up width $> 5/4 + 12\delta$. But, as the width of any rectangle in A is at least $1/4 - 4\delta$, we do not have sufficient total width to pack 3 rectangles from A. This proves the claim.

   Hence there is exactly one rectangle from B'. Call it $b'_{l'}$. Since $b'_{l'}$ can only overlap vertically with b_l it must be that $h(b_l) + h(b'_{l'}) \le 1$, and hence that $w(b_l) + w(b'_{l'}) \ge 1/2$. Moreover, by Observation 1, $w(b_l) + h(b_l) + h(b'_{l'}) + w(b'_{l'}) = 3/2$. Now, by summing Equations 2.1,2.2 and 2.3 and using Observation 1 we have that each inequality in Equations 2.1,2.2 and 2.3 is satisfied with equality. To complete the argument, suppose b_l intersects line L_1, let a_1, a_2, a_3 denote the rectangles in $A \cup A'$ which also intersect L_1. Thus, we have that $w(a_1)+w(a_2)+w(a_3)+w(b_l) = 1$. None of the a_i, $1 \le i \le 3$ can lie in A', because otherwise it is always the case that $w(a_1) + w(a_2) + w(a_3) + w(b_l) > 1$. Since all a_1, a_2 and a_3 lie in A, by Lemma 2.1 it follows that these rectangles correspond to a triple $t_l = (x_i, y_j, z_k)$.

THEOREM 2.1. *There is no Asymptotic PTAS for the 2-Dimensional Bin Packing Problem unless $P = NP$.*

Proof. If the MAX-3-DM problem has a matching consisting of q tuples, then we can get a bin packing solution which uses $|T|$ bins as follows. For each of the tuples in the matching, create a good bin as described in Lemma 2.3. For each t_l not in the matching, we put b_l and b'_l along with a dummy rectangle, and hence we use $q + (|T| - q) = |T| \le 3q$ bins.

Assume now that every feasible solution of the MAX-3-DM problem has at most $(1 - \varepsilon)q$ triples. We will show that any solution to the corresponding bin packing problem uses at least $(1 + \epsilon/33)|T|$ bins. Consider any feasible solution to the bin-packing instance. There will be exactly $n_d = |T| - q$ bins with dummy objects. Lemma 2.3 implies that if a bin is not good then it either has at most 7 rectangles or else it has at most 3 rectangles from $A \cup B$. Let n_g denote the number of good bins. Since the set of good bins corresponds to some feasible solution by Lemma 2.3 we have $n_g \le (1 - \varepsilon)q$. Among the bins which are not good let n_{b_1} denote the number of bins (other than the bins with dummy objects) which contain at most 7 rectangles and let n_{b_2} denote the rest of the bins, (note that these are precisely the bins that have eight rectangles but 3 or fewer rectangles from $A \cup B$).

Since any solution must cover all the rectangles in $A \cup B$ and any bin with a dummy rectangle can have at most one rectangle from $A \cup B$, we have the constraint that

$$4n_g + 4n_{b_1} + 3n_{b_2} + n_d \geq 3q + |T|$$

Equivalently,

$$4n_g + 4n_{b_1} + 3n_{b_2} \geq 4q$$

Similarly, since all the rectangles must be covered, we have that

$$8n_g + 7n_{b_1} + 8n_{b_2} + 2n_d \geq 6q + 2|T|$$

Equivalently,

$$8n_g + 7n_{b_1} + 8n_{b_2} \geq 8q$$

Adding the equations above, $12n_g + 11n_{b_1} + 11n_{b_2} \geq 12q$. Equivalently, $n_g + n_{b_1} + n_{b_2} \geq 12q/11 - n_g/11$. Adding the bins with dummy objects, this implies that the total number of bins used is at least $|T| - q + 12q/11 - n_g/11 = |T| + (q - n_g)/11 \geq |T| + \varepsilon q/11 \geq |T|(1 + \epsilon/33)$.

Now if there is an APTAS for 2-dimensional bin packing, then for every $\epsilon > 0$, there exists an algorithm A_ϵ and a constant c_ϵ, such that for instances I if $|Opt(I)| > c_\epsilon$, then $A_\epsilon \leq (1 + 2\epsilon)|Opt(I)|$. Thus for any $\epsilon > 0$, if $q > c_\epsilon$ we can distinguish between two instances of the MAX-3-DM problem with $|T'| = q$ and $|T'| \leq (1 - 66\varepsilon)q$, which is an NP-hard problem by [17]

3 PTAS for square packing

In this section we give a PTAS for the special case where we want to pack squares into bins.

We begin with some notation and definitions: The problem instance I is a collection of squares specified by their sizes. For any collection of squares C, we use $A(C)$ to denote the total area of the squares in C, and $OPT(C)$ is used to denote the optimum number of unit size bins used to pack the squares in C.

Before describing our overall algorithm, we first describe a subroutine (known as the Decreasing height shelf algorithm) that we use to pack squares in an arbitrary rectangular region. A *shelf* is a row of items having their bases on a line which is either the base of the bin or the line drawn at the top of the tallest item packed in the shelf below.

Given a collection $\mathcal{C}$ such that the square indexed by i has size $s_i \times s_i$ and $s_1 \geq s_2 \geq \ldots \geq s_n$ which are to be packed in a rectangular region R, the *Decreasing Height Shelf* algorithm is defined as follows:

Starting with a packing of s_1 in the bottom left corner of R, continue packing objects $s_2, \ldots, s_{i_1}$ at the base of R such that s_{i_1+1} does not fit. At this point, we close this shelf with a vertical line L_1 at height s_1. We open a new shelf with line L_1 as the base, and continue packing objects $s_{i_1+1}, \ldots, s_{i_2}$ and so on. If after k shelves, the item s_{i_k+1} does not fit in the shelf above (i.e. we do not have enough space to begin the $(k+1)^{th}$ shelf, we close the rectangle R and do not pack any more objects in it.

This algorithm was described and studied first in [5]. The following is a crucial property of the decreasing height shelf algorithm.

LEMMA 3.1. *Let C_s be an arbitrary collection of squares, where each square has size at most s. Let R be a rectangular region of size $a \times b$ (means width a and height b), in which C_s is to be packed. Assume that the collection C_s is large enough that it never runs out of squares while packing R. Then, packing the squares in C_s according to the decreasing height shelf heuristic wastes at most $s(a+b)$ area.*

Proof. Let s_i denote the size of job starting shelf i, and l_i denote size of the last job in shelf i. The waste at the end of this shelf is at most $s_{i+1}s_i$ (because s_{i+1} could not fit in this shelf. The waste at the top of each shelf (excluding the area already accounted for) is at most $(s_i - l_i)(a - s_{i+1})$.

Suppose there are k shelves. The waste in the region where we could not begin the $k+1^{th}$ shelf is at most $(l_k)a$.

Adding up all this waste we get, $l_k a + \sum_{i=1}^{k}(s_i - l_i)a + l_i s_{i+1}$. Observing that $s_i \geq l_i \geq s_{i+1}$, this waste is upper bounded by $s_1 a + \sum_{i=1}^{k} l_i s_1 \leq s_1(a+b)$

COROLLARY 3.1. *While packing squares in a unit square bin, Decreasing height shelf heuristic is a 6 approximation with respect to the area.*

Proof. Suppose the largest square has size at least $\sqrt{2}-1$, then we pack at least $(\sqrt{2}-1)^2 > 1/6$ of the area. If the largest square is smaller than $\sqrt{2}-1$, by the above lemma we waste at most $2(\sqrt{2}-1)$ and hence use at least $3 - 2\sqrt{2} > 1/6$.

Let $\epsilon_0 = 1 > \epsilon_1 > \epsilon_2 > \ldots > \epsilon_k = 0$ be a sequence of numbers, where $k = \lceil 1/\epsilon \rceil$ and $\epsilon_i = \epsilon^{2^i - 1}$. Note that $\epsilon_i = \epsilon\epsilon_{i-1}^2$, for $1 \leq i \leq k-1$.

Let I_i denote the subcollection of squares in I whose sizes lie in the range $[\epsilon_{i-1}, \epsilon_i)$, for $1 \leq i \leq k$. Observe that the I_i partition I into a collection of k disjoint sets. Thus, there exists some m such that $A(I_m) \leq \epsilon A(I)$. We call all squares in I_m as medium, squares in I_i such that $i < m$ are called large, and squares in I_i for $i > m$ are called small. This definition is due to Sevastianov and Woeginger [19] and is very helpful in design of PTASes for scheduling and packing problems.

We now describe our algorithm. The algorithm will have 3 phases: In the first phase we pack the medium squares. In the second phase we find a close to optimum packing of the large squares, and in the final phase we pack the small squares. We now describe each of these phases.

Packing medium squares:

Take the medium squares and pack them in separate bins using the Decreasing height shelf algorithm. These bins are closed and will never be used again.

Packing large squares:

Let L denote the set of large objects. Let $l = |L|$ and $s_1 \geq \ldots \geq s_l$ denote the sizes of the squares in L. We form $g = 1/\epsilon_m$ groups as follows. The first group L_1, consists for largest $\lceil l/g \rceil$ pieces the second of the next largest $\lceil l/g \rceil$ and so on. Construct a new instance L' obtained by discarding the first group and for each other group, rounding the size of the square to the size of the largest square in this group. Note that $OPT(L') \leq OPT(L)$. This rounding trick is due to Fernandez de la Vega and Lueker [8] and is very popular in bin packing literature.

Our algorithm packs each square in L_1 in its own separate bin. To obtain a packing of $L_2 \cup \ldots \cup L_g$, we will obtain a packing of L'. Clearly, a valid packing of L' can be used to pack the squares in $L_2 \cup \ldots \cup L_g$.

Notice that the instance L' has at most g distinct job sizes, call them $s'_1, \ldots, s'_g$. We now show how to obtain a close to optimum packing of L'. The ideas are similar to that of one dimensional bin packing [8].

Consider the ways in which a single bin can be packed. These can be described by a k-tuple $t = (t_1, \ldots, t_g)$, where t_i denote the number of squares of size s'_i in the bin. Since each large square has size at least ϵ_{m-1}, the total number of such squares in any bin can be at most $1/\epsilon^2_{m-1}$. Thus, the number of all possible tuples can be at most $\binom{g + 1/\epsilon^2_{m-1}}{g} \leq 2^{g+1/\epsilon^2_{m-1}} \leq 2^{2g}$ which is a constant.

Call a tuple $T = (t_1, \ldots, t_g)$ feasible, if there is a way to pack all the objects corresponding to t in a single bin. It is easy to check if a tuple is feasible in constant time. For every square packed in a bin we can assume that it is shifted to leftmost and bottom most position which implies that each of the lower left corner coordinates of this square can be represented as a sum of constant number of square sizes. The algorithm first figures out in constant time the set of all feasible tuples. Let $\mathcal{T} = \{T_1, \ldots, T_q\}$ denote the set of all feasible tuples. Let T_{ij} denote the number of squares of size t_j in T_i.

We form a linear program (denoted by LP) as follows:

$$\min \sum_{i=1}^{q} x_i \tag{3.4}$$

subject to

$$\begin{aligned} \sum_{i=1}^{q} T_{ij} x_i &\geq n_j, && \text{for each } j = 1, \ldots, g \\ x_i &\geq 0, && \text{for each } i = 1, \ldots, q \end{aligned}$$

The variable x_i corresponds to the (fractional) number of bins which have configuration corresponding to the tuple T_i. Observing that any basic optimal solution to this LP has at most g non-zero variables x_i. We can obtain an integral solution by simply rounding each non-zero x_i to next integer. This increases the cost by at most g. After that we define x_i packings corresponding to the tuple i this guarantees that for every size type there is enough squares used in that packing and therefore all large squares can be packed.

Packing Small Squares:

Consider the bins containing the large objects in an arbitrary order. For each bin we first divide the gaps into rectangular shapes using the following procedure:

Fix a bin B. For each square in B extend its top side and bottom side horizontally in both directions until it hits another square or the boundary of the bin B. This divides the gaps in the bin into rectangular regions. See figure 2. Moreover, the number of such rectangular regions is at most $4/\epsilon^2_{m-1} + 1$. This follows by adding the lines one by one. Since there are at most $1/\epsilon^2_{m-1}$ squares in a bin, we have at most $4/\epsilon^2_{m-1}$ such lines and each line adds at most one additional rectangular region.

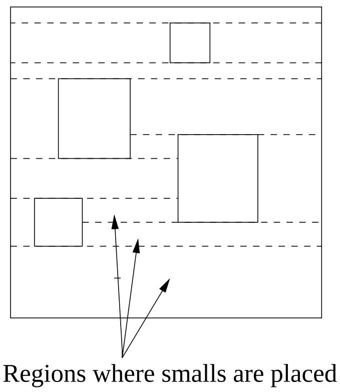

Figure 2: *Regions where small squares are packed*

To pack the small objects, consider the rectangular regions in any arbitrary order, and pack the small objects in these regions according to the decreasing height shelf algorithm.

If all the small objects cannot be packed in the rectangular regions, open additional new bins to pack the remaining small objects. Again, the small objects in the new bins are packed according to the decreasing height shelf algorithm.

THEOREM 3.1. *The algorithm described above is an asymptotic PTAS for square packing in two dimensions.*

Proof. By Corollary 3.1, the number of bins N_m used for the medium sized squares is at most $6A(I_m)$, since $A(I_m) \leq \epsilon A(I)$, we have that

$$N_m \leq 6A(I_m) \leq 6\epsilon A(I) \leq 6\epsilon OPT(I) \tag{3.5}$$

We now account for the number of bins used by the large squares. Since each large square has size at least ϵ_{m-1} and hence area at least ϵ_{m-1}^2, we have that $OPT(I) \geq OPT(L) \geq l\epsilon_{m-1}^2$.

The number of bins used by our algorithm to pack the objects in L_1 is

$$\lceil l/g \rceil \leq \epsilon_m l + 1 = \epsilon\epsilon_{m-1}^2 l + 1 \leq \epsilon OPT(I) + 1 \tag{3.6}$$

Next, as $LP(L') \leq OPT(L')$, $IP(L') \leq LP(L') + g$ and as $OPT(L') \leq OPT(L) \leq OPT(I)$ we have that

$$IP(L') \leq OPT(I) + g \tag{3.7}$$

By Equations 3.6 and 3.7 it follows that the total number of bins used by our algorithm to pack the large objects is at most

$$(1+\epsilon)OPT(I) + g + 1 \tag{3.8}$$

If no additional bins are opened for the small squares, the result follows from Equations 3.5 and 3.8.

If additional bins need to be opened for the small squares, consider the total amount of area wasted in the bins containing the large squares.

Note that the size of the small squares is at most ϵ_m, and hence by Lemma 3.1 the total area in any rectangular region of dimensions $a \times b$ is at most $\epsilon_m(a+b) \leq 2\epsilon_m$. Since there were at most $8/\epsilon_{m-1}^2$ such regions that each such bin has waste at most

$$2\epsilon_m \cdot 8/\epsilon_{m-1}^2 \leq 16\epsilon$$

Similarly, the area wasted in the bins in which only small squares are packed (expect for one bin) is trivially upper bounded by 16ϵ.

Thus the total number of bins used in this case is

$$\begin{aligned} A(I)/(1-16\epsilon) + 1 + N_m &\leq \\ A(I)/(1-16\epsilon) + 1 + 6\epsilon A(I) &\leq \\ A(I)(1+O(\epsilon)) + 1 & \end{aligned}$$

for ϵ sufficiently small. Thus the result follows.

Extension to the $d-$dimensional case: The above algorithm can extends directly to yield an APTAS in the $d-$dimensional case for fixed d. The only difference is that we choose $\epsilon_0 = 1$ and $\epsilon_i = \epsilon\epsilon_{i-1}^d$ for $1 \leq i \leq k-1$ and $k = \lceil 1/\epsilon \rceil$. It is easy to see that the algorithm for $d = 2$ described above gives an APTAS for this case.

References

[1] A. Caprara. Packing 2-dimensional bins in harmony. In *Foundations of Computer Science*, pages 490–499, 2002.

[2] C. Chekuri and S. Khanna. On multi-dimensional packing problems. In *Symposium on Discrete Algorithms (SODA)*, pages 185–194, 1999.

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

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

[5] E. G. Coffman, M. R. Garey, D. S. Johnson and R. E. Tarjan. Performance bounds for level-oriented two dimensional packing algorithms. SIAM J. Computing 9 (1980), pages 808-826.

[6] J. R. Correa and C. Kenyon. Approximation schemes for multidimensional packing. this proceedings.

[7] J. Crisik and G. Woeginger. On-line packing and covering problems. In *Online Algorithms: The State of the Art, editors A. Fiat and G. Woeginger*, pages 147–177, 1998.

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

[9] L. Epstein and R. Van Stee. Optimal online bounded space multidimensional packing. In *CWI Technical Report Number SEN-R0301*, 2003.

[10] C. E. Ferreira, F. K. Miyazawa, and Y. Wakabayashi. Packing squares into squares. *Pesquisa Operacional*, 19(2):223–237, 1999.

[11] K. Jansen and G. Zhang. On rectangle packing: maximizing benefits. this proceedings.

[12] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37:27–35, 1991.

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

[14] Claire Kenyon and Eric Remila. Approximate strip packing. In *Foundations of Computer Science (FOCS)*, pages 31–36, 1996.

[15] Y. Kohayakawa, F.K. Miyazawa, P. Raghavan, and Y. Wakabayashi. Multidimensional cube packing. In

Brazilian Symposium on Graphs, Algorithms and Combinatorics, 2001.

[16] J. Y. T. Leung, T. W. Tam, C. S. Wong, G. H. Young and F. Y. L. Chin. Packing Squares into a Square. *Journal of Parallel and Distributed Computing*, 10: 271-275, 1990.

[17] E. Petrank. The hardness of approximation: gap location. *Computational Complexity*, 4:133–157, 1994.

[18] S. Seiden and R. van Stee. New bounds for multidimensional packing. *Algorithmica*, 36(3):261–293, 2003.

[19] S. Sevastianov and G. Woeginger. Makespan minimization in open shops: a polynomial time approximation scheme. *Networks and matroids; Sequencing and scheduling. Math. Programming*, 82, no. 1-2, Ser. B:191–198, 1998.

[20] G. Woeginger. There is no asymptotic PTAS for two-dimensional vector packing. *Information Processing Letters*, 64:293–297, 1997.