



Resource augmentation in two-dimensional packing with orthogonal rotations[☆]

José R. Correa

School of Business, Universidad Adolfo Ibáñez, Av. Presidente Errázuriz 3485, Las Condes, Santiago, Chile

Received 15 December 2004; accepted 20 February 2005

Available online 31 May 2005

Abstract

We consider the problem of packing two-dimensional rectangles into the minimum number of unit squares, when 90° rotations are allowed. Our main contribution is a polynomial-time algorithm for packing rectangles into at most OPT bins whose sides have length $(1 + \varepsilon)$, for any positive ε . Additionally, we show near-optimal packing results for a number of related packing problems.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Bin packing; Approximation algorithms; Polynomial time approximation schemes

1. Introduction

The classical *two-dimensional bin-packing* problem consists of placing rectangles of specified size into the minimum number of rectangular bins, in such a way that no two rectangles overlap. The most well-studied version of the problem is the case in which the rectangles cannot be rotated and each rectangle must be packed parallel to the edges of a bin. However, in many applications one needs to consider the problem where rotations—in particular 90° rotations—are allowed.

Several variants of this two-dimensional packing problem have been considered in the last 50 years. An important one among them is the problem of finding the minimum area rectangle in which a given set of rectangles can be placed. This problem, known as *minimum rectangle packing*, is particularly relevant when *orthogonal rotations* (i.e., 90° rotations) are allowed. Indeed, Murata et al. [16] state that:

The earliest and the most critical stage in VLSI layout design is the placement. The background of which is the rectangle packing problem: Given a set of rectangular modules of arbitrary sizes, place them without overlap on a plane within a rectangle of minimum area.

Closely related to minimum rectangle packing is the two-dimensional (and three-dimensional)

[☆] An extended abstract appeared in the proceedings of the Latin American Conference in Graphs Algorithms and Combinatorics, Santiago, Chile, August 2004. This work has been partially supported by Grant MECESUP UCH 0109.

E-mail address: correa@uai.cl.

strip-packing problem. Here, we are again given a set of rectangles (or boxes in the three-dimensional case) and a strip of fixed width (resp. fixed basis) and infinite height. The goal is to find a placement of the rectangles (resp. boxes) of minimum total height; depending on the application, rotations may or may not be allowed. A variant of the three-dimensional strip-packing problem with rotations is the so-called *z-oriented three-dimensional packing* problem, in which orthogonal rotations are allowed only on the z -axis. Strip-packing problems have many industrial applications, particularly in stock cutting. Moreover, applications in computer science are also very common. For instance, the z -oriented three-dimensional packing problem was introduced by Li and Cheng [12] as a model for job scheduling in partitionable mesh connected systems.

1.1. Assumptions

A standard assumption in the design of algorithms for rectangle-packing problems, which we also adopt here, is that the given rectangles (items) are of bounded size. Also, note that for bin-packing problems without rotations we can always assume (by rescaling) that the bins are unit squares. However, this is not true in two-dimensional bin packing under 90° rotations. For simplicity, we present the results in this paper when the bins are unit squares, nevertheless, as we essentially use exhaustive search to deal with rotations, our results hold in the general case as well. The same comment applies to the basis of the strip in strip-packing problems.

1.2. Previous work

Leung et al. [11] proved that given a collection of squares it is NP-Hard to decide in polynomial time whether these can be packed into a single bin. Later, Ferreira et al. [8] noted that this implies that no $2 - \varepsilon$ approximation algorithm for two-dimensional bin packing (with or without rotations) can exist unless $P = NP$. In the case without rotations, Caprara [4] exhibited an algorithm with an asymptotic performance ratio of $1.691\dots$, improving the $2 + \varepsilon$ obtained implicitly from the asymptotic polynomial-time approximation scheme (APTAS for short) for strip packing by Kenyon and Rémila [10]. On the negative side, Bansal and Sviridenko [3] recently showed that there

is no APTAS for two-dimensional bin packing unless $P = NP$. They, also designed an APTAS for the special case in which the rectangles are squares. This last result was independently obtained by Correa and Kenyon [6], who, in addition, obtained an algorithm that finds a packing using no more than OPT bins of size $(1 + \varepsilon) \times (1 + \varepsilon)$. The reader is referred to [2] for more details on these results. Under 90° rotations, however, the situation is less clear: No hardness of approximation result is known, while, when the bins are unit squares, the best approximation algorithm is due to Epstein and Van Stee [7] has a performance guarantee of 2.25, and, in the general case, the best known is a 2.64-approximation algorithm, due to Miyazawa and Wakabayashi [15].

In the context of minimum rectangle packing without rotations, only heuristic approaches had been proposed. However, a polynomial-time approximation scheme (PTAS) was recently obtained [6]. Prior to this work, the situation for rectangle packing with rotations was still open. In fact, we are not aware of any theoretical approximation result.

The breakthrough result for two-dimensional strip packing without rotations was obtained by Kenyon and Rémila [10] who gave an APTAS, improving an old 1.25 approximation by Baker et al. [1]. In the three-dimensional case however, Miyazawa and Wakabayashi [13] obtained the best-known algorithm, that has an asymptotic approximation ratio between 2.5 and 2.67. Moreover, as noted by Kenyon [9], the results in [3] imply that no APTAS for the problem can exist unless $P = NP$.

For two-dimensional strip packing with 90° rotations the best-known algorithm has a performance guarantee of 1.5, while in the three-dimensional case with rotations 2.25 is currently the best known when the strip has unit width [7]. The bounds respectively become 1.613 and 2.76 in the general case [13]. For the z -oriented three-dimensional packing problem, Miyazawa and Wakabayashi [14] proved that an algorithm with asymptotic performance ratio of α can be translated into an α -asymptotic approximation algorithm for the three-dimensional strip packing problem without rotations. They also give an algorithm with an asymptotic performance guarantee between 2.5 and 2.67 (actually this is how they derived the performance bound for three-dimensional strip packing). Later, Epstein and van Stee gave a 2.25 asymptotic

approximation algorithm for the problem, which they call, the “This Side Up” problem [7]. It is important to mention that all results in [7] hold for bins of unit size, while those in [13–15] hold for general bins. Therefore, their results are not directly comparable.

1.3. Our results

In this short paper, we obtain results for packing problems allowing orthogonal rotations. Our main result, proved in Section 2, reads as follows:

Theorem 1.1. *There exists an algorithm A which, given a list I of n rectangles and a positive number ε , produces a packing of I into $A(I)$ copies of $[0, 1 + \varepsilon]^2$ such that*

$$A(I) \leq \text{OPT}(I),$$

where $\text{OPT}(I)$ is the minimum number of unit squares in which I can be packed. The running time of A is polynomial in n for fixed ε .

To prove this result we borrow ideas from [6], where the problem without rotations was considered. Similar to that case, our algorithm relies on an appropriate partition of rectangles into “large”, “small”, “flat”, and “medium”, in such a way that the medium rectangles are negligible. Large rectangles are both tall and wide while flat rectangles are long in one dimension but very small on the other. Then, we round up the long sides of rectangles, i.e., the long side of flat rectangles and both sides of large rectangles. With this rounding, rectangles which are large are reduced to a constant number of types, and thus they are packed by exhaustive search. Small rectangles are packed using the Next-Fit Decreasing Height (NFDH) shelf heuristic; described at the beginning of Section 2. The main difficulty is to deal with flat rectangles. In particular, in the case with orthogonal rotations, we need to guess a right partition of flat rectangles into “vertical” and “horizontal”. To overcome this problem, we note that, after rounding, essentially there is only a constant number of flat rectangles which need a significantly different treatment. This is true since two equally wide flat rectangles are treated similarly by the strip packing algorithm of Kenyon and Rémila [10]. We can, therefore, show that it is enough to look

at a polynomial number of partitions of flat rectangles to guess a solution that is sufficiently close to the optimal.

In Section 3 we first apply the ideas behind Theorem 1.1 to obtain similar augmented PTASs for other two-dimensional packing problems. Indeed, we obtain a PTAS for the minimum rectangle-packing problem with rotations, and an ε -augmented asymptotic PTAS for two-dimensional strip packing with rotations. Furthermore, we will combine these ideas, the ideas in [6], and the linear programming approach of Kenyon and Rémila to obtain the following results: an APTAS for three-dimensional strip packing without rotations when the boxes have a square basis; an ε -augmented asymptotic PTAS for three-dimensional strip packing without rotations; and an ε -augmented asymptotic PTAS for the z -oriented three-dimensional packing problem. The first two of these facts were also observed by Kenyon [9].

2. Two-dimensional bin packing with rotations

The algorithm presented here is built upon the algorithm described by Correa and Kenyon [6]. Indeed, after decomposing the input into “large”, “small” and “flat” rectangles, we use exhaustive search to find a packing of large rectangles and NFDH to pack small rectangles. As we shall see, these two types of rectangles can still be packed almost optimally in the case with rotations. The key difficulty is to decide which flat rectangles should be placed horizontally and which vertically. To this end we find a way of looking at only a polynomial number of partitions of flat rectangles into “horizontal” and “vertical”, and we prove that at least one of these is sufficiently close to the optimal one.

An important subroutine of our algorithm is the NFDH heuristic, defined by Coffman et al. [5]. In the NFDH heuristic, rectangles are first sorted in non-increasing order of their height. Then, they are packed left-justified on a level (starting from the bottom) until there is insufficient space at the right to accommodate the next rectangle. At that point, the next level is defined by drawing a horizontal line through the top of the first (and therefore largest) rectangle placed in the previous level.

2.1. The algorithm

The following algorithm is strongly based on that in [6]. The main difference is that, as rotations are allowed we need to do many more iterations through the main loop. For clarity, we present the algorithm in full detail. The algorithm packs any list I of rectangles into no more than $\text{OPT}(I)$ bins of size $(1 + 15\varepsilon) \times (1 + 15\varepsilon)$, where $\text{OPT}(I)$ is the minimum number of unit squares in which I can be packed. To obtain the exact result in Theorem 1.1 it is enough to reassign $\varepsilon \leftarrow \varepsilon/15$ and apply the algorithm to this newly defined ε .

Algorithm for rectangle packing with 90° rotations

Input. Let I denote the input list consisting of n rectangles to be packed. Since rotation is allowed, assume that the i th rectangle has width a_i and height b_i , with $0 < b_i \leq a_i \leq 1$. Denote also by $\text{Sf}(I) = \sum_{i=1}^n a_i b_i$, the total surface of the input.

Partitioning the input. For $j \in \{1, 2, \dots, 2/\varepsilon\}$, let M_j denote those rectangles i such that $a_i \in (\varepsilon^{2(j+1)+1}, \varepsilon^{2j+1}]$ or $b_i \in (\varepsilon^{2(j+1)+1}, \varepsilon^{2j+1}]$. Let j_0 be such that the total surface area of the rectangles of M_{j_0} is minimum. Let $\varepsilon' = \varepsilon^{2j_0+1}$, and define the partition $I = M_{j_0} \cup L \cup F \cup S$, where

- $L = \{i : a_i > \varepsilon' \text{ and } b_i > \varepsilon'\}$
- $S = \{i : a_i < \varepsilon'\varepsilon^2 \text{ and } b_i < \varepsilon'\varepsilon^2\}$
- $F = \{i : a_i > \varepsilon' \text{ and } b_i < \varepsilon'\varepsilon^2\}$

Rounding the input. We now round every coordinate greater than ε' up to the nearest multiple of $\varepsilon'\varepsilon$. Denote by I' the rounded instance, i.e. I' is $L \cup F$ after rounding.

Let C denote the number of distinct rectangles in L : thus L contains ℓ_i rectangles of type i , for $1 \leq i \leq C$.

Rounding and partitioning the output. We define bin types by decomposing $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$ squares as follows:

- We consider all possible packings of (large) rectangles of type i , $1 \leq i \leq C$, into a $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$

square, such that the corners of the rectangles have coordinates which are integer multiples of $\varepsilon'\varepsilon$. Note that this should include considering rotations of rectangles in L . That is, in each possible packing, a rectangle in L is associated with its type ($1 \leq i \leq C$) and with an orientation (either horizontal or vertical).

- For each possible packing of large rectangles into a $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$ square, we consider the area of the bin which is still uncovered as a union of small square cells of side length $\varepsilon'\varepsilon$ (where each square is positioned at integer multiples of $\varepsilon'\varepsilon$), and label each cell either H or V . Together with this labelling of the uncovered area, this packing of large rectangles defines a bin type. Let K be the total number of bin types.

Consider all rectangles in F after rounding. Let $R = \lceil (1 + \varepsilon)/\varepsilon'\varepsilon \rceil$ be the maximum number of possible widths of rectangles in F . For each integer $1 \leq r \leq R$ (i.e., $r\varepsilon'\varepsilon$ is less than $1 + \varepsilon$) consider all rectangles in F whose width is $r\varepsilon'\varepsilon$ and number them (in arbitrary order) $1, 2, \dots, t_r$ (of course, $t_r \leq n$). Thus, after rounding, each rectangle in F is uniquely determined by its width and a positive integer.

Main loop. For each choice of m_1, \dots, m_R with $m_r \leq t_r$, $r = 1, \dots, R$, and for each (n_1, \dots, n_K) such that $\sum_j n_j \leq n$, we attempt to construct a packing of $I' \cup S$ using n_j bins of type j . The packing satisfies the following: the rectangles from L are packed in the spaces reserved for them in the bin type; the cells labelled H are only used to horizontally place rectangles from F whose associated integer is at most m_r , where $r\varepsilon'\varepsilon$ is the width of the rectangle, and rectangles from S ; the cells labelled V are only used to vertically place rectangles from F whose associated integer is more than m_r , where $r\varepsilon'\varepsilon$ is the width of the rectangle, and rectangles from S . In the remainder of the algorithm we let H (resp. V) be the set of rectangles in F that we will attempt to pack in cells labelled H (resp. V); thus, $F = H \cup V$. The packing is constructed as follows:

- (1) To decide whether the rectangles from L can be placed, we check that for every rectangle type i , $1 \leq i \leq C$, the number ℓ_i of type i rectangles in I'

is less than or equal to the total space available for them:

$$\ell_i \leq \sum_{j=1}^K n_j \left(\begin{array}{l} \text{number of type } i \text{ rectangles} \\ \text{positioned in type } j \text{ bins} \end{array} \right).$$

Here we look at type i rectangles positioned in type j bins, either horizontally or vertically.

(2) We use the following algorithm for packing the rectangles from H :

(a) For each bin type j , consider the union U of the cells labelled H . Drawing horizontal lines at y -coordinates integer multiples of $\varepsilon'\varepsilon$, we can interpret U as a union of horizontal strips of height $\varepsilon'\varepsilon$ and width multiple of $\varepsilon'\varepsilon$. For each integer multiple ℓ of $\varepsilon'\varepsilon$, let $h_\ell^{(j)}$ denote the sum of the heights of the strips of width ℓ in a type j bin. Let h_ℓ denote the total height of the strips of width ℓ in the packing which we are currently constructing,

$$h_\ell = \sum_{1 \leq j \leq K} n_j h_\ell^{(j)}.$$

(b) We now consider the problem of packing rectangles from H with rounded widths into two-dimensional bins of height h_ℓ and width ℓ . Note that the number of different width types for rectangles in H is bounded above by a constant. To do this we will solve the following fractional strip-packing problem. Consider all configurations (w_1, w_2, \dots) of widths (including the empty configuration) which are multiples of $\varepsilon'\varepsilon$ and sum to at most $1 + \varepsilon$. The number of such configurations is constant. Let A_{ir} denote the number of occurrences of width $i\varepsilon'\varepsilon$ in configuration r . Let B_i denote the sum of all heights of the rectangles of H whose width equals $i\varepsilon'\varepsilon$. We define one variable $x_r^{(\ell)}$ for each strip width ℓ and for each configuration r whose widths sum to at most ℓ . We find, in polynomial time, a basic feasible solution to the following system of linear constraints, if it

exists.

$$(\forall i) \quad B_i \leq \sum_{\ell, r} x_r^{(\ell)} A_{ir},$$

$$(\forall \ell) \quad \sum_r x_r^{(\ell)} \leq h_\ell,$$

$$(\forall r, \ell) \quad x_r^{(\ell)} \geq 0.$$

(c) We place rectangles from H in the configurations thus defined, proceeding in a greedy fashion.

(d) We cut back bins of height h_ℓ into strips of height $\varepsilon'\varepsilon$ and place them back into the bins. If a rectangle is cut we throw this rectangle away from the packing and pack it later.

(3) Similarly, we pack the rectangles of V into the parts of the bins labelled V .

Let $M \subseteq H$ denote the set of rectangles which either did not fit in the fractional packing after (c) or are cut in the process (d). Analogously, we define the set of rectangles $M' \subseteq V$ that remained unpacked after step (3).

(4) We choose an arbitrary orientation for the rectangles in S and pack them into the $\varepsilon'\varepsilon \times \varepsilon'\varepsilon$ cells which have available space, using the NFDH algorithm.

(5) We expand each bin by adding 13 thin $1 \times \varepsilon$ horizontal strips and also 13 thin $\varepsilon \times 1$ vertical strips and use them to pack the rectangles from $M_{j_0} \cup M \cup M'$ using an $O(1)$ -approximation algorithm such as NFDH in the horizontal strips and Next Fit Decreasing Width (NFDW) in the vertical strips.

Output. We output the best packing among all feasible packings of I thus constructed.

2.2. Analysis of the running time

We show that, for fixed $\varepsilon > 0$, the algorithm just described runs in time polynomial in n .

Theorem 2.1. *Given a list I of n rectangles, our algorithm runs in time polynomial in n for fixed $\varepsilon > 0$.*

Proof. Let us first see that the input can be partitioned and rounded efficiently. Indeed, as $j_0 \leq 2/\varepsilon$, we have that $\varepsilon' \geq \varepsilon^{4/\varepsilon} = \Omega(1)$. Thus, partitioning the input

requires $O(n)$ operations, the same as rounding the input.

To bound the number of bin types, note that the number C of large rectangle types (after rounding) is at most $(1/\varepsilon'\varepsilon)^2 = O(1)$. A bin type can be defined by labelling each $\varepsilon'\varepsilon \times \varepsilon'\varepsilon$ cell by $H, V, (i, V)$ or (i, H) , for some $1 \leq i \leq C$. Label (i, V) (resp. (i, H)) means that in such cell a rectangle of type i will be placed vertically (resp. horizontally). Thus, the number K of bin types is at most $(2C + 2)^{(1+\varepsilon)/(\varepsilon'\varepsilon)^2} = O(1)$.

Let us now look at the number of iterations of the main loop. Clearly, $R = (1 + \varepsilon)/\varepsilon'\varepsilon \leq (1 + \varepsilon)/\varepsilon^{1+4/\varepsilon} = O(1)$ and m_1, \dots, m_R (i.e., the partition of F into H and V) can be chosen in at most n^R ways. Also, the number of choices of n_1, \dots, n_K is at most n^K . Therefore, the number of iterations through the main loop is at most $n^{(K+R)}$ which is polynomial in n .

Finally, let us see what happens within an iteration of the main loop. Step (1) requires $O(1)$ arithmetic operations with numbers of size $O(n)$. In step (2), the number of strip widths is at most $1/\varepsilon'\varepsilon = O(1)$ while the number of configurations is at most $2^{2/(\varepsilon'\varepsilon)} = O(1)$. Multiplying, the number of variables in the system of linear constraints is $O(1)$. The number of constraints is at most $2/(\varepsilon'\varepsilon) = O(1)$. The coefficients $A_{i,r}$ are bounded by $O(1)$ and B_i are written on at most $O(\log n)$ bits. Hence the linear program (and thus steps (2) and (3)) can be solved efficiently in polylogarithmic time. The NFDH and NFDW algorithms runs in time $O(n \log(n))$. Therefore, steps (4) and (5) require time $O(n \log(n))$.

Overall, we conclude that the algorithm runs in time

$$O(n^{K+R} n \log(n)) = O\left(n^{2^{\tilde{O}(1/\varepsilon)}}\right),$$

which is polynomial in n for fixed ε . \square

2.3. Analysis of correctness

The analysis of the algorithm borrows several ideas from Section 3 in [6]. Therefore, in several proofs we will refer the reader to that paper.

Let I' denote the rounded input. It was noted in [6] that a packing of I into B unit size bins, can be converted into a packing of $I' \cup S \cup M_{j_0}$ into B bins of size $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$, such that any rectangle with width at least ε' is positioned at an x -coordinate which

is an integer multiple of $\varepsilon'\varepsilon$, and any rectangle with height at least ε' is positioned at a y -coordinate which is an integer multiple of $\varepsilon'\varepsilon$. Here, the width and height of a rectangle correspond to those quantities using the orientation of the current packing.

The following lemma says that once we have chosen the orientation of the rectangles (i.e., defined the partition $F = H \cup V$), a cell cannot be shared by an horizontal and a vertical rectangle.

Lemma 2.2 (6, Lemma 3.3). *Consider a packing of I' into bins of size $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$, satisfying the conditions above. Consider any cell $\mathcal{C} = [m\varepsilon'\varepsilon, (m + 1)\varepsilon'\varepsilon] \times [p\varepsilon'\varepsilon, (p + 1)\varepsilon'\varepsilon]$ in any bin. Then either $H \cap \mathcal{C} = \emptyset$ or $V \cap \mathcal{C} = \emptyset$.*

Lemma 2.3. *The sets M_{j_0}, M and M' satisfy $\text{Sf}(M_{j_0}) \leq \varepsilon \text{Sf}(I)$, $\text{Sf}(M) \leq 5\varepsilon \text{Sf}(I)$ and $\text{Sf}(M') \leq 5\varepsilon \text{Sf}(I)$ respectively. Moreover, all rectangles in $M_{j_0} \cup M \cup M'$ fit into the thin strips added along the bins in step (5).*

Proof. Each rectangle in I belongs to at most two sets M_j . Therefore

$$\frac{2}{\varepsilon} \text{Sf}(M_{j_0}) \leq \sum_{1 \leq j \leq 2/\varepsilon} \text{Sf}(M_j) \leq 2 \text{Sf}(I),$$

and then $\text{Sf}(M_{j_0}) \leq \varepsilon \text{Sf}(I)$. The fact that $\text{Sf}(M) \leq 5\varepsilon \text{Sf}(I)$ and $\text{Sf}(M') \leq 5\varepsilon \text{Sf}(I)$ is analogous to Lemma 3.4 in [6] (see [2] for details).

Let us see that all rectangles in $M_{j_0} \cup M \cup M'$ fit into the strips added along the bins in step (5). The total surface of $M_{j_0} \cup M \cup M'$ is no more than $11\varepsilon \text{Sf}(I) \leq 11\varepsilon \text{OPT}(I)$. Furthermore, $M_{j_0} \cup M \cup M'$ can be partitioned (and rotated) into two sets, A and B such that: A contains only rectangles with $a_i < \varepsilon' < \varepsilon^2$ and $\text{Sf}(A) \leq 6\varepsilon \text{Sf}(I)$; and B contains only rectangles with $b_i < \varepsilon' < \varepsilon^2$ and $\text{Sf}(B) \leq 6\varepsilon \text{Sf}(I)$.

As all rectangles in A have width smaller than ε^2 , for small enough ε , NFDW packs A into the $13 \times \text{OPT}(I)$ added strips of size $\varepsilon \times 1$. On the other hand, NFDH does the work for the rectangles in B . Note that the number of thin strips that we need to add to the $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$ bins depends on the algorithm used to pack rectangles in $M_{j_0} \cup M \cup M'$. \square

The following lemma is a slight variation of a result due to Coffman et al. (cf. Theorem 3 in [5]). It simply says that NFDH wastes a negligible amount of space when packing only small rectangles.

Lemma 2.4. *Consider the two-dimensional NFDH heuristic applied to rectangles in S . When NFDH cannot place any other rectangle in a bin of size $a \times b$ then the total unused area in that bin is no more than $(\varepsilon' \varepsilon^2)(a + b)$.*

We are now ready to give the overall analysis of the algorithm.

Theorem 2.5. *Given a list I of n rectangles and a positive number ε , our algorithm produces a packing of I into $A(I)$ copies of $[0, 1 + 15 \cdot \varepsilon]^2$ such that*

$$A(I) \leq \text{OPT}(I),$$

where $\text{OPT}(I)$ is the minimum number of unit squares in which I can be packed, with 90° rotations allowed.

Proof. Consider an input list I and let I' be the rounded input. We know that there exists a packing of $I' \cup S \cup M_{j_0}$ into no more than $\text{OPT}(I)$ bins of size $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$ such that

- (i) Rectangles are oriented in the same way as in an optimal packing of I .
- (ii) In the current orientation, any rectangle with width at least ε' is positioned at an x -coordinate which is an integer multiple of $\varepsilon'\varepsilon$, and any rectangle with height at least ε' is positioned at a y -coordinate which is an integer multiple of $\varepsilon'\varepsilon$.

Consider then such a packing for $I' \cup S \cup M_{j_0}$ and let H and V be respectively, the rectangles in F (defined as in the “Partitioning the Input” step) which are placed horizontally and vertically. By Lemma 2.2 we know that in such a packing all cells \mathcal{C} intersect either a rectangle in L , H or V but not two of them. In other words in such a packing of $I' \cup S \cup M_{j_0}$ each cell is labelled either V , H , (i, V) or (i, H) for $i = 1, \dots, C$; or will have no label at all. Let us call this labelling the optimal labelling.

Clearly, our algorithm will eventually guess a labelling of the cells that coincides with the optimal labelling. At the same time, our algorithm will guess a

partition of F into H and V which need not be optimal. Let us see that at least one of the partitions of F our algorithm will guess is sufficiently close to the optimal. To this end, let $H_l^* \subset H$ denote the set of width l horizontal rectangles in the (rounded) optimal packing. Let $H_l \subset H$ denote the set of width l rectangles in the best possible partition the algorithm can guess. By best possible we mean the partition minimizing the maximum over l of the total height difference between H_l^* and H_l , i.e., minimizing

$$\max_{\text{widths } l} \left| \sum_{i \in H_l^*} b_i - \sum_{i \in H_l} b_i \right|,$$

where b_i is the height of rectangle i in the current orientation. Note that H_l^* and H_l may be very different sets. However, since all rectangles in F are of height at most $\varepsilon'\varepsilon^2$, in the best possible partition of F into H and V the algorithm can guess, the total height of rectangles in H_l^* and that of rectangles in H_l will differ by at most $\varepsilon'\varepsilon^2$, for all widths l . Of course, the analogous condition will simultaneously hold for rectangles in V .

After guessing the right partition of F into horizontal and vertical rectangles, the algorithm will find a feasible packing of all rectangles in L and almost all rectangles in H and V . This is done in steps (2) and (3). By Lemma 2.3, the unpacked rectangles M_{j_0} , M and M' are of small surface and they are packed in the extra space added in step (5) of the algorithm.

It remains to see that the small rectangles, S , will be successfully packed by NFDH in step (4) using the remaining space available in the $\text{OPT}(I)$ bins of size $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$. Assume, by contradiction, that a new bin is indeed opened in step (4). At this step, we distinguish four types of $\varepsilon'\varepsilon \times \varepsilon'\varepsilon$ cells: the ones completely filled with a rectangle in L , the ones filled with only rectangles in S , the ones filled only with rectangles in H or V , and the ones partly filled with rectangles in H or V and partly with rectangles in S . By Lemma 2.4, all cells are almost filled. Indeed, the wasted space in each cell is at most $\varepsilon'\varepsilon^2 \times (\varepsilon'\varepsilon + \varepsilon'\varepsilon) = 2 \times (\varepsilon')^2 \varepsilon^3$, while the total area of each cell is exactly $(\varepsilon'\varepsilon)^2$. Thus a fraction $(1 - 2\varepsilon)$ of each cell’s surface is filled. Overall, this implies that a fraction $(1 - 2\varepsilon)$ of the first $\text{OPT}(I)$ bins is filled, and then the total surface that has been filled in the first $\text{OPT}(I)$

bins of size $(1 + 2\varepsilon) \times (1 + 2\varepsilon)$ squares is at least $(1 - 2\varepsilon)\text{OPT}(I)(1 + 2\varepsilon)^2 \geq \text{OPT}(I)$, for $\varepsilon < 1/4$. \square

3. Further results on packing with rotations

To conclude, we give some notes on other packing problems with 90° rotations in which the ideas of Section 2 apply. The first such problem is the minimum rectangle packing problem.

Theorem 3.1. *There exists an algorithm A which, given a list I of n rectangles and a positive number ε , produces a packing of I into a rectangle $R(I)$ such that*

$$\text{Sf}(R(I)) \leq (1 + \varepsilon)\text{OPT}(I),$$

where $\text{OPT}(I)$ is the area of the smallest rectangle in which I can be packed (where rotations are allowed) and $\text{Sf}(r)$ denotes the area of rectangle r . The running time of A is polynomial in n for fixed ε .

To obtain this result, it is enough to note that if I denotes an instance of total surface $\text{Sf}(I)$, the algorithm in Section 2 can be easily adapted to solve the underlying approximate decision problem: Given n rectangles with sides $a_i, b_i \in (0, 1]$, a rectangle R of size $a \times b$, and $\delta > 0$: determine if all n rectangles can be packed (with 90° rotations) in a rectangle of size $[(1 + \delta)a + \delta] \times [(1 + \delta)b + \delta]$ or they cannot be packed in R . As done in [6], if for at least one rectangle $i \in I$ both a_i and b_i are not too small, say at least ε , we can solve the decision problem (taking $\delta = \varepsilon^2$) for a polynomial number of rectangles R to obtain Theorem 3.1 (since we know that NFDH can always pack I in a square of size $2\text{Sf}(I) \times 2\text{Sf}(I)$). Otherwise, either all a_i 's or all b_i 's are smaller than ε (note that orthogonal rotations are allowed). In this case, the result can be obtained by a surface argument using a shelf packing heuristic (adapting a result in [5]).

Very similarly, this approximate decision problem can be used to obtain an approximately optimal packing for the two-dimensional strip packing problem under 90° rotations.

Theorem 3.2. *There exists an algorithm A which, given a list I of n rectangles of height bounded by h , and a positive number ε , produces a packing of I into a*

strip of dimensions $(1 + \varepsilon) \times ((1 + \varepsilon)\text{OPT}(I) + O(h/\varepsilon^2))$ whenever a packing of I into a strip of dimensions $1 \times \text{OPT}(I)$ exists. The running time of A is polynomial in n for fixed ε .

However, an even stronger result for this latter problem may hold. In fact, it was conjectured in [10] that two-dimensional strip packing admits an APTAS. To the best of our knowledge, this remains open.

For three-dimensional strip packing without rotations, already the results in [6], combined with the linear programming technique in [10], have some interesting consequences. First, they imply the existence of a polynomial-time algorithm A , which, given a list I of boxes with square basis and height bounded by h , and a positive number ε , produces a packing of I into a strip of dimensions $1 \times 1 \times ((1 + \varepsilon)\text{OPT}(I) + O(h/\varepsilon^2))$ whenever a packing of I into a strip of dimensions $1 \times 1 \times \text{OPT}(I)$ exists. This APTAS represents an improvement over the prior best-known algorithm, that has an asymptotic performance ratio of 2.36 [14]. Secondly, one can obtain a near-optimal resource-augmented packing result for the general case. In other words, one can prove that there exists an algorithm A which, given a list I of boxes of height bounded by h and a positive number ε , produces a packing of I into a strip of dimensions $(1 + \varepsilon) \times (1 + \varepsilon) \times ((1 + \varepsilon)\text{OPT}(I) + O(h/\varepsilon^2))$ whenever a packing of I into a strip of dimensions $1 \times 1 \times \text{OPT}(I)$ exists. The key idea behind these two results is that in both cases we only need to consider a constant number of different basis configurations. Then, the linear program of Kenyon and Rémila [10] will handle the problem by associating one variable for each such configuration. In an optimal LP solution the value of that variable will correspond to the total height for which the associated configuration has to be used in an optimal “fractional” packing. The solution can finally be rounded with the help of three-dimensional versions of NFDH.

Essentially the same technique can be applied in the context of the z -oriented three-dimensional packing problem (in which rotations are allowed only on the z -axis). The ideas in Section 2 allow us to consider a constant number of basis configurations, thus, they may again be combined with the linear program approach in [10] to obtain the following approximation result.

Theorem 3.3. *For the z -oriented three-dimensional packing problem there exists an algorithm A which, given a list I of n boxes of height bounded by h and a positive number ε , produces a packing of I into a strip of dimensions $(1 + \varepsilon) \times (1 + \varepsilon) \times ((1 + \varepsilon)\text{OPT}(I) + O(h/\varepsilon^2))$ whenever a packing of I into a strip of dimension $1 \times 1 \times \text{OPT}(I)$ exists. The running time of A is polynomial in n for fixed ε .*

This result is actually the best possible in some sense. Indeed, the reduction of Miyazawa and Wakabayashi [14] (which says that an α -approximation algorithm for z -oriented three-dimensional packing can be transformed into an α -approximation algorithm for three-dimensional strip packing) together with the hardness result of Bansal and Sviridenko [3] imply that no APTAS for the z -oriented three-dimensional packing problem exists unless $P = NP$.

Acknowledgements

The author thanks Claire Kenyon for introducing him to multidimensional packing problems and, specifically, for asking the question about rotations. The paper greatly benefited from insightful discussions with Claire and from the useful comments of Michael Wagner and an anonymous referee.

References

- [1] B.S. Baker, D.J. Brown, H.P. Kasteff, A $5/4$ algorithm for two-dimensional packing, *J. Algorithms* 2 (1981) 348–368.
- [2] N. Bansal, J.R. Correa, C. Kenyon, M. Sviridenko, Bin packing in multiple dimensions: inapproximability results and approximation schemes, manuscript, November 2004.
- [3] N. Bansal, M. Sviridenko, New approximability and inapproximability results for 2-dimensional bin packing, *Proceedings of the 15th Symposium on Discrete Algorithms (SODA)*, 2004, pp. 189–196.
- [4] A. Caprara, Packing 2-dimensional bins in harmony, *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002, pp. 490–499.
- [5] E.G. Coffman, M.R. Garey, D.S. Johnson, R.E. Tarjan, Performance bounds for level-oriented two dimensional packing algorithms, *SIAM J. Comput.* 9 (1980) 808–826.
- [6] J.R. Correa, C. Kenyon, Approximation schemes for multidimensional packing, *Proceedings of the 15th Symposium on Discrete Algorithms (SODA)*, 2004, 179–188.
- [7] L. Epstein, R. van Stee, This side up!, *Proceeding of the 2nd Workshop on Approximation and Online Algorithms (WAOA)*, 2004, to appear.
- [8] C.E. Ferreira, F.K. Miyazawa, Y. Wakabayashi, Packing squares into squares, *Pesquisa Operacional* 19 (1999) 223–237.
- [9] C. Kenyon, Personal communication, October 2003.
- [10] C. Kenyon, E. Remila, A near optimal solution to a two-dimensional cutting stock problem, *Math. Oper. Res.* 25 (2000) 645–656.
- [11] J.Y.T. Leung, T.W. Tam, C.S. Wong, G.H. Young, F.Y.L. Chin, Packing squares into a square, *J. Parallel Distributed Comput.* 10 (1990) 271–275.
- [12] K. Li, K.-H. Cheng, Static job scheduling in partitionable mesh connected systems, *J. Parallel Distributed Comput.* 10 (1990) 152–159.
- [13] F.K. Miyazawa, Y. Wakabayashi, An algorithm for the three-dimensional packing problem with asymptotic performance analysis, *Algorithmica* 18 (1997) 122–144.
- [14] F.K. Miyazawa, Y. Wakabayashi, Approximation algorithms for the orthogonal z -oriented 3-D packing problem, *SIAM J. Comput.* 29 (2000) 1008–1029.
- [15] F.K. Miyazawa, Y. Wakabayashi, Packing problems with orthogonal rotations, *Proceedings of the 6th Latin American Symposium on Theoretical Informatics (LATIN)*, Lecture Notes on Computer Science, vol. 2976, Springer, Berlin, 2004, pp. 359–368.
- [16] H. Murata, K. Fujiyoshi, S. Nakatake, Y. Kajitani, VLSI module placement based on rectangle-packing by the sequence-pair, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems* 15 (1996) 1518–1524.