A near-optimal solution to a two-dimensional cutting stock problem

Claire Kenyon
LRI, bat 490,Université Paris Sud
91405 Orsay Cedex, France.
Email: Claire.Kenyon@lri.fr

Eric Rémila
Grima, IUT Roanne (Université J. Monnet)
20 avenue de Paris, 42334 Roanne, France.
and
LIP, ENS-Lyon, CNRS URA 1398,
46, allée d'Italie, 69364 Lyon Cedex 07, France.
Email: Eric.Remila@ens-lyon.fr

Abstract

We present an asymptotic fully polynomial approximation scheme for strip-packing, or packing rectangles into a rectangle of fixed width and minimum height, a classical NP-hard cutting-stock problem. The algorithm finds a packing of n rectangles whose total height is within a factor of $(1+\epsilon)$ of optimal (up to an additive term), and has running time polynomial both in n and in $1/\epsilon$. It is based on a reduction to fractional bin-packing.

1 Introduction

1.1 Results

We consider the following version of a two-dimensional cutting stock problem: given a supply of material consisting of one rectangular strip of fixed width 1 and large height, given a demand of n rectangles of widths and heights in the interval $[0, 1]$, the problem is to cut the strip into the demand rectangles while minimizing the waste, i.e. minimizing the total height used.

This is called the strip-packing problem, a natural generalization of bin-packing to two dimensions. We do not allow the demand rectangles to be rotated (in many applications, rotations are not allowed because of constraints such as the patterns of the cloth or of the grain of the wood). In computer science, strip-packing models the scheduling of independent tasks, each requiring a certain number of contiguous processors or memory locations for a certain length of time; the width of the strip represents the total number of processors or memory locations available, and the height represents the completion time.

Strip-packing is NP-hard, since it includes bin-packing as a special case (when all heights are equal). Thus, unless $P = NP$, one cannot find an efficient algorithm for constructing the optimal packing. One then seeks to design approximate heuristics A with good performance guarantees.

Definition 1 : *Let $A(L)$ denote the height used by A on input L, and $Opt(L)$ denotes the height used by the optimal algorithm on input L. The absolute performance ratio of A is $\sup_L A(L)/Opt(L)$ The asymptotic performance ratio of A is $\limsup_{Opt(L)\to\infty} A(L)/Opt(L)$.*

In this paper, we focus on the asymptotic performance ratio. Our main result is the following :

Theorem 1 : *There is an algorithm A which, given a list L of n rectangles whose side lengths are at most 1, and a positive number ϵ, produces a packing of L in a strip of width 1 and height $A(L)$ such that :*

$$A(L) \leq (1+\epsilon)Opt(L) + O(1/\epsilon^2).$$

The time complexity of A is polynomial in n and $1/\epsilon$.

In other words, the paper presents a fully asymptotic fully polynomial time approximation scheme for strip-packing.

The running time is $O(n \log n + \log^3 n\epsilon^{-6} \log^3 \epsilon^{-1})$ if we use a subroutine from [13]. It may be possible to improve on this by using further ideas from [11] and [17].

Two-dimensional stock-cutting with stages goes as far back from 1965 with work by Gilmore and Gomory [9]. In computer science, many ideas for strip-packing originally arose from bin-packing studies. In 1980, Baker, Coffman and Rivest showed that the "Bottom-Left" heuristic has asymptotic performance ratio equal to 3 when the rectangles are sorted by decreasing widths [2]. Coffman, Garey, Johnson and Tarjan studied algorithms where the rectangles are placed on "shelves" using one-dimensional bin-packing heuristics, and showed that the First-Fit shelf algorithm has asymptotic performance ratio of 2.7 when the rectangles are sorted by decreasing height (this defines the First-Fit-Decreasing-Height algorithm) [4]. The asymptotic performance ratio of the best heuristic was further reduced to 2.5 [15], then to 4/3 [10] and finally to 5/4 [1]. The absolute performance ratio has also been the object of much research, with the best current algorithm having a performance of ratio 2 [16] [14].

In 1991, de la Vega and Zissimopoulos used a very different approach, based on a reduction to integer linear programming, to design a $(1+\epsilon)$ asymptotic approximation scheme for strip packing, in the case when all rectangle widths and heights are bounded below and above by constants [8]. In other words, they solve the strip-packing problem as long as the rectangles are neither too flat nor too narrow. Their work was inspired by approximation schemes developed for one-dimensional bin-packing, based on linear programming. This direction was explored by de la Vega and Lueker in 1981 (with a reduction of bin-packing to constant-size integer linear programming) [7] and later by Karp and Karmarkar, to yield a fully asymptotic polynomial time approximation scheme for bin-packing by reduction to fractional bin-packing [11]. In order to compare our algorithm with the one developed in [8], one must note that the algorithm in [8], though linear-time in terms of the number of rectangles, is worse than exponential in terms of ϵ, thus inherently impractical.

In this paper (which is an extended version of [12]), we use many ideas from [7], [8], [11].

1.2 Methods

Bin-packing and strip-packing are closely related, and many ideas which originally arose from bin-packing can also be applied to strip-packing. It is thus natural to try to extend the linear programming approach from bin-packing [7], [11] to strip-packing.

One obstacle to such an extension comes from the small input items (rectangles of small width or height), since both approximation schemes developed in [7], [10] for bin-packing first set small input values aside, then construct an efficient packing of the other values, and finally add the small values in a greedy way so as to form a packing which is still efficient.

In the case of strip-packing, however, there is no efficient way in general to complete a packing of a strip (which may have many little gaps of odd shapes), when adding the rectangles of small width or of small height.

One should note that rectangles of small width are not inherently difficult: In the extreme case when all input rectangles have small width, the First-Fit-Decreasing-Height shelf heuristic (FFDH) is very efficient.

Theorem 2 [4] *If all rectangles of L have width less than or equal to $1/k$, then the height $FFDH(L)$ achieved by the FFDH heuristic on list L satisfies*

$$FFDH(L) \leq s(L)(1 + 1/k) + 1,$$

where $s(L)$ is the total area of the rectangles in L.

Our original hope was that the method of [8] and the FFDH heuristics might be combined to get an efficient approximation scheme for general strip-packing. The integer linear program devised in [8] can unfortunately not be combined with FFDH, but another linear programming approach, which *only constructs packings with few odd-shaped gaps*, works.

The algorithm is structured as follows: first, the rectangles are divided into two sublists: the "narrow" rectangles, i.e. whose widths are less than a positive constant ϵ', and the "wide" rectangles (i.e. which are not narrow). We take the wide rectangles and change their widths by using a variation of Karmakar and Karp's linear rounding, so as to build another list L_{sup} of wide rectangles with only a *bounded number of distinct widths*. Relaxing the constraints to allow horizontal cuts of the rectangles, we obtain a *fractional bin-packing problem*, defined in the next section, from which (either by [10] or by [13]) we deduce a strip-packing for L_{sup} which is close to optimal. Finally, the packing produced has a specific "nice" shape which makes the insertion of narrow rectangles possible, while still keeping the packing close to optimal.

For stock-cutting applications, where machines can only perform edge-to-edge cuts parallel to the strip's length or width, called "guillotine cuts", it is worthwhile to remark that our algorithm is also applicable to guillotine cuts. In fact, it can be realized by 5 stages of consecutive parallel (and, consequently, permutable) guillotine cuts. This will be explained further at the time of the presentation of the algorithm.

2 The Algorithm

2.1 Definitions

A rectangle is given by its width w_i and height h_i, with $0 < w_i, h_i \leq 1$. The area (resp. height) of a list $L = ((w_1, h_1), (w_2, h_2), ..., (w_n, h_n))$ of rectangles is the sum of the areas (resp. heights) of the rectangles of L. We assume that the list is ordered by non-increasing widths: $w_1 \geq w_2 \geq ... \geq w_n$.

Remark : The assumption that the heights are less than 1 is also made in several other papers as [4]. Without it, one could scale the items arbitrarily and thus the absolute and the asymptotic performance ratios would coincide, so that there would be no hope of getting a fully polynomial asymptotic scheme.

A strip-packing of a list L of rectangles is a positioning of the rectangles of L within the vertical strip $[0, 1] \times [0, +\infty)$, so that all rectangles have disjoint interiors. If rectangle (w_i, h_i) is positioned at $[x, x + w_i] \times [y, y + h_i]$, then y is called the lower boundary and $(y + h_i)$ the upper boundary of the rectangle. The height of a strip-packing is the uppermost boundary of any rectangle. Let $Opt(L)$ denote the minimum height of a strip-packing of L: $Opt(L) = \inf\{\text{height of } f \text{ such that } f \text{ is a packing of } L\}$.

A fractional strip-packing of L is a packing of any list L' obtained from L by subdividing some of its rectangles by horizontal cuts: each rectangle (w_i, h_i) is replaced by a sequence $(w_i, h_{i_1}), (w_i, h_{i_2}), ..., (w_i, h_{i_{k_i}})$ of rectangles such that $h_i = \sum_j h_{i_j}$.

We first present the algorithm when the number of distinct widths of the rectangles is bounded by some value m, and all widths are larger than some constant ϵ'.

2.2 Reduction from the "few and wide" case to fractional strip-packing

Throughout this subsection, we assume that the n rectangles of L only have m distinct widths, $w'_1 > w'_2 > .... > w'_m > \epsilon'$, where the exact values of m and of

ϵ' will be given later.

This section contains one main new idea of the paper, which is a reduction from this special case of strip-packing to fractional strip-packing.

To the input L, we associate a set of *configurations*. A configuration is defined as a multi-set of widths (chosen among the m widths) which sum to less than 1 (i.e. capable of occurring at the same level). Their sum is called the *width* of the configuration.Without loss of generality, the configurations can be assumed to be ordered by non increasing widths.

Let q be the number of distinct configurations, and let α_{ij} denote the number of occurrences of width w_i' in configuration C_j.

To each (possibly fractional) strip-packing of L of height h, we associate a vector $(x_1, \ldots, x_q)$, $x_i \geq 0$, in the following manner. Scan the packing bottom-up with a horizontal sweep line $y = a$, $0 \leq a \leq h$. Each such line is canonically associated to a configuration $(\alpha_1, ..., \alpha_m)$, where α_i is the number of rectangles of width w_i' whose interior is intersected by the sweep line. Let x_j, $1 \leq j \leq q$, denote the measure of the a's such that the sweep line $y = a$ is associated to configuration C_j. For example, let A denote the rectangle $3/7 \times 1$ and B denote the rectangle $2/7 \times 3/4$, and assume that the input L consists of three rectangles of type A and four rectangles of type B. There are seven configurations, listed below.

	configuration	α_{1j} =number of A's	α_{2j} =number of B's
C_1	$3/7, 2/7, 2/7$	1	2
C_2	$3/7, 3/7$	2	0
C_3	$2/7, 2/7, 2/7$	0	3
C_4	$3/7, 2/7$	1	1
C_5	$2/7, 2/7$	0	2
C_6	$3/7$	1	0
C_7	$2/7$	0	1

The vector corresponding to the strip-packing figure 1 is $(3/2, 5/4, 0, 0, 0, 0, 0)$.

The fractional strip-packing problem is canonically defined as follows : given a list L of rectangles, construct a fractional strip packing of minimal height.

Lemma 1 *The fractional strip-packing problem is the linear program:*

$$\textit{minimize } (1.x) \textit{ subject to } x \geq 0 \textit{ and } Ax \geq B,$$

where 1 is the all-ones vector, A is the $m \times q$ matrix $(\alpha_{ij})_{1 \leq i \leq m, 1 \leq j \leq q}$, and $B = (\beta_1, ..., \beta_m)$, β_i denoting the sum of the heights of all rectangles of width w_i'.

Proof: Consider the vector $(x_1, \ldots, x_q)$ associated to a fractional strip-packing of L of height h. We clearly have: $x_j \geq 0$, $\sum_j x_j = h$, and for every i, $\sum_j \alpha_{ij} x_j = \beta_i$.

Conversely, to each vector $(x_1, \ldots, x_q)$ such that $x_i \geq 0$ and $\sum_j \alpha_{ij} x_j = \beta_i$,we can associate a fractional strip-packing of L of height $\sum_j x_j$ in the following manner.

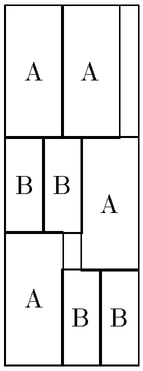

Figure 1: A strip-packing of L.

Partition the strip of width 1 and height $\sum_j x_j$ into j pieces of width 1 and heights x_j $(1 \leq j \leq q)$. In the jth piece, for each i such that $\alpha_{ij} \neq 0$, draw α_{ij} columns of width w'_i and height x_j. Finally, for each i, fill up the columns of width w'_i with the input rectangles of width w'_i in a greedy manner, cutting the rectangles as you go so as to fill each column exactly up to height x_j. This works perfectly since $\sum_j \alpha_{ij} x_j$ is greater than or equal to $\geq \beta_i$, the total height of the rectangles of width w'_i. We have constructed a fractional strip-packing of L of height $\sum_j x_j$.

$\diamond$

We now recall the fractional bin-packing problem studied in [10]. In this problem, the input is a set of n items of m different types, i.e. which take only m distinct sizes in $(\epsilon, 1]$. A configuration is a multi-set of types which sum to at most 1 (i.e. capable of being packed within a bin). If q denotes the number of configurations, then a feasible solution to the fractional bin-packing problem is a vector $(x_1, \ldots, x_q)$ of non-negative numbers such that if α_{ij} is the number of pieces of type i occurring in configuration j, then for every i, $\sum_j \alpha_{ij} x_j$ is at least equal to the number of input pieces of type i. The goal is to minimize $\sum_j x_j$. Let z be the minimum. The fractional bin-packing problem with tolerance t has for its goal to find a feasible solution such that $\sum_j x_j \leq z + t$, and was solved in [10] in polynomial time.

Thus, fractional strip-packing is precisely fractional bin-packing.

Theorem 3 [*11*] *There exists an algorithm for fractional bin-packing with additive tolerance t, such that if n is the number of items and m the number of distinct items, then the running time is polynomial in $m, n, 1/\epsilon$ and $1/t$ and if m' denotes the number of non-zero coordinates of the solution given by the algorithm, we have $m' \leq m$*

One should note that the running time, though polynomial, tends to be of large degree, and thus is inefficient in practice. However in [13] the running time is reduced to $O(n \log n + \log^3 n\epsilon^{-6} \log^3 \epsilon^{-1})$.

We now relate fractional strip-packing to strip-packing.

Lemma 2 *If L has a fractional strip-packing $(x_1, \ldots, x_q)$ of height h and with at most m non-zero x_j's, then L has an (integral) strip-packing of height at most $h + m$.*

Proof: Consider a fractional strip-packing $(x_1, ..., x_q)$ of L, of height $\sum_i x_i = h$, and with at most m non-zero coordinates x_i's. Up to renaming, we assume that the non-zero coordinates are $x_1, ..., x_{m'}$, with $m' \leq m$. Let h_{max} be the maximum height of any rectangle of L. We construct a strip packing of L of height $h + mh_{max}$ in the following way.

We fill in the strip bottom-up, taking each configuration in turn. Let $x_j > 0$ denote the variable corresponding to the current configuration. Configuration j will be used between level $l_j = (x_1 + h_{max}) + \ldots + (x_{j-1} + h_{max})$ and level $l_{j+1} = l_j + x_j + h_{max}$ (initially $l_1 = 0$). For each i such that $\alpha_{ij} \neq 0$, we draw α_{ij} columns of width w'_i going from level l_j to level l_{j+1}. After this is done for all j's, we take all the columns of width w'_i one by one in some arbitrary order, and and fill them in with the rectangles of width w'_i in a greedy manner (some small amount of space may be wasted on top of each column).

We claim that all the rectangles fit. The proof is by contradiction. Assume a rectangle R, of width w'_i, does not fit in any column of width w'_i. Consider such a column. It has height $x_j + h_{max}$, for some x_j, while R has height at most h_{max}. Since R does not fit, the column must be filled up to more than x_j. Summing over all columns of width w'_i, we get that the cumulative heights of all the rectangles placed so far in these columns is more than $\sum_j \alpha_{ij} x_j$, which is $\geq \beta_i$, a contradiction.

This proves that the construction yields a strip-packing of L. Its height is $(x_1 + h_{max}) + \ldots + (x_{m'} + h_{max}) = h + mh_{max}$, and h_{max} is at most 1, hence the lemma.

$\diamond$

This gives a straightforward algorithm for strip-packing in the special case studied in this section.
1) Solve fractional strip packing on L with tolerance 1, as in [10] (the solution has at most m non-zero coordinates).
2) From the fractional strip packing, construct a strip packing of L as in the proof of the lemma above.

Moreover, a crucial point for the sequel (i. e. for the addition of narrow rectangles) is that this strip packing leaves some well-structured free space.

Important remark : structure of a layer (see figure 2)

Let $c_1 \geq c_2 \geq ... \geq c_{m'}$ denote the widths of the m' configurations used above. The layer $[0, 1] \times [l_i, l_{i+1}]$ can be divided into three rectangles:
i) the rectangle $R_i = [c_i, 1] \times [l_i, l_{i+1} - 1]$, which is completely free and will later

be used to place the narrow rectangles,
ii) the rectangle $R'_i = [0, c_i] \times [l_i, l_{i+1} - 1]$, which is completely filled by wide rectangles,
iii) the rectangle $R''_i = [0, 1] \times [l_{i+1} - 1, l_{i+1}]$, which is partially filled in some complicated way by wide rectangles overlapping from R_i, and whose free space is now considered as wasted space, and will not be used in the remainder of the construction.

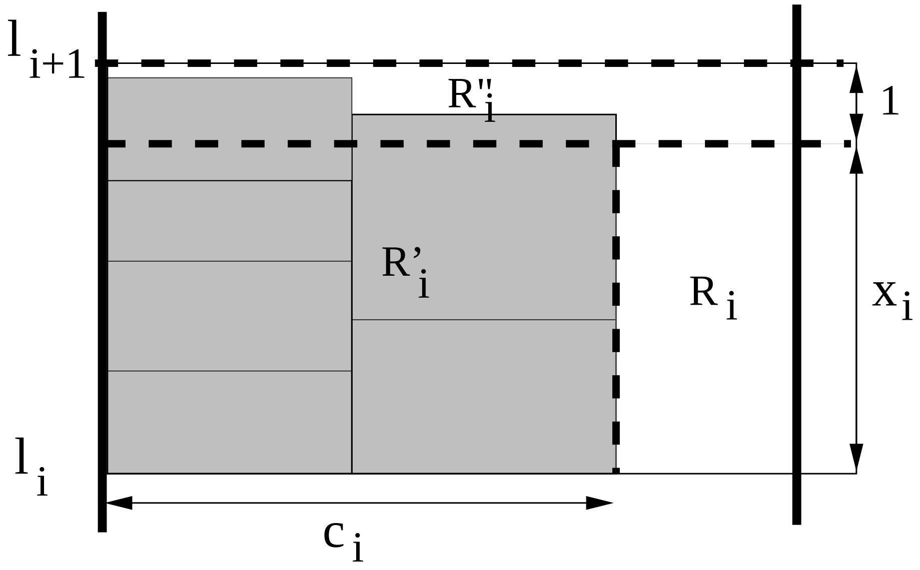

Figure 2: structure of a layer

2.3 Reduction from general strip-packing to the "few and wide" case

In the general case, we have a list $L_{general}$ with many distinct widths, some of which may be arbitrarily small. We use appropriate extensions of two ideas of [7]: elimination of small pieces, and grouping. The purpose of elimination is to insure all rectangles are wider than some ϵ'. The purpose of grouping is to insure that the number of distinct widths is bounded.

2.3.1 Elimination of narrow rectangles.

During the elimination phase, we partition the list $L_{general}$ into two sublists: L_{narrow}, containing all the rectangles of width at most ϵ', and L, containing all the rectangles of width larger than ϵ'. During the next stage, we will focus on L.

2.3.2 Grouping.

This is the other main idea of the paper.

We define a partial order on lists of rectangles by saying that $L \leq L'$ if there is an injection from L to L' such that each rectangle of L has smaller width and height than the associated rectangle of L'.

Given a list L of rectangles whose widths are larger than ϵ', we will now approximate L by a list L_{sup} such that $L \leq L_{sup}$, and the rectangles of L_{sup} only have m distinct widths.

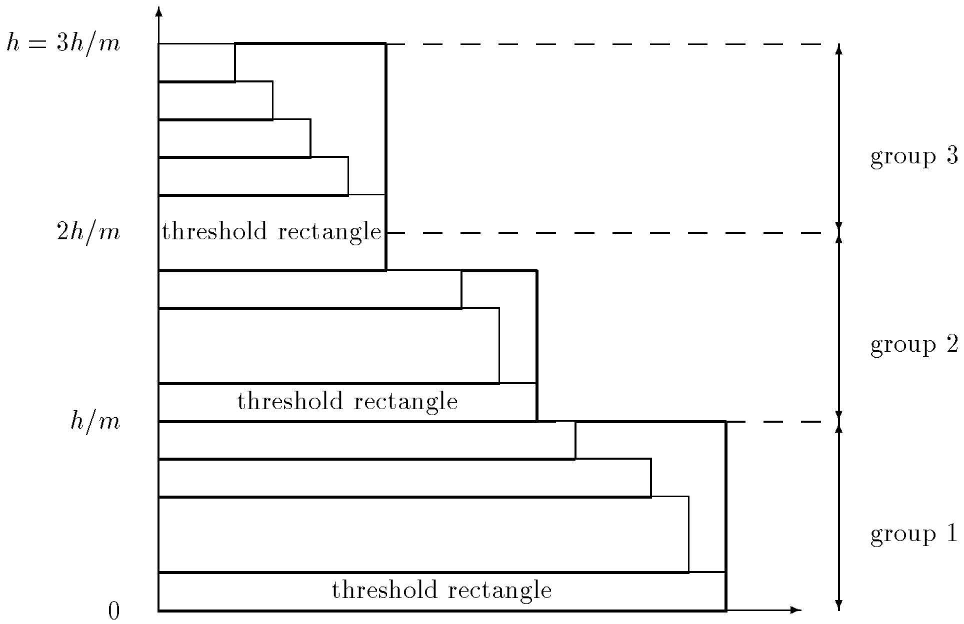

Figure 3: grouping the rectangles, example when $m = 3$. The thick lines show how to extend the rectangles to construct L_{sup}.

In order to define L_{sup}, we first stack up all the rectangles of L by order of non-increasing widths, to obtain a left-justified stack of total height $h(L)$. We define $(m-1)$ threshold rectangles, where a rectangle is a threshold rectangle if its interior or lower boundary intersects some line $y = ih(L)/m$, for some i between 1 and $m-1$ (see example figure 3). The threshold rectangles separate the remaining rectangles into m groups. The widths of the rectangles in the first group are then rounded up to 1, and the widths of the rectangles in each subsequent group are then rounded up to the width of the threshold rectangle below their group. This defines L_{sup}. Note that if all rectangle heights are equal, this is exactly the linear grouping defined in [7], and thus this can be seen as an extension of [7]. Also note that L_{sup} consists of rectangles which have only m distinct widths, all greater than ϵ'.

We construct a strip-packing of L_{sup} using the ideas of section 2.2. A packing of L is trivially deduced by using the relation $L \leq L_{sup}$ and placing each rectangle of L inside the position of the associated rectangle of L_{sup}.

To get a packing of $L_{general}$, the narrow rectangles must now be added.

2.3.3 Adding the narrow rectangles.

Order the rectangles of L_{narrow} by decreasing heights. We add the rectangles of L_{narrow} to the current strip-packing, trying to use the m' free rectangular areas $R_1, R_2, ..., R_{m'}$ as much as possible, according to a "Modified Next-Fit-Decreasing-Height" algorithm as follows. Use the Next-Fit-Decreasing-Height (NFDH) heuristic to pack rectangles in R_1: in this heuristic, the rectangles are packed so as to form a sequence of sublevels. The first sublevel is simply the bottom line. Each subsequent sublevel is defined by a horizontal line drawn through the top of the first (and hence highest) rectangle placed on the previous sublevel. Rectangles are packed in a left-justified greedy manner, until there is insufficient space to the right to accommodate the next rectangle ; at that point, the current sublevel is discontinued, the next sublevel is defined and packing proceeds on the new sublevel.

When a new sublevel cannot be started in R_1, start the next sublevel at the bottom left corner of R_2 using NFDH again, and so on until $R_{m'}$. When a rectangle cannot be packed in $R_1, ...$ or $R_{m'}$, use NFDH to pack the remaining rectangles in the strip of width 1 starting above $R_{m'}$, at level $l_{m'+1}$. This gives a packing of $L_{general}$.

We are now ready to summarize the overall algorithm.

2.4 The overall algorithm

Parameters: ϵ' (the threshold narrow/wide) and m (the number of groups). We set $\epsilon' = \epsilon/(2+\epsilon)$ and $m = (1/\epsilon')^2$.
Input: a list of rectangles $L_{general}$.
1) Perform the partition $L_{general} = L_{narrow} \cup L$ to set aside the rectangles of width less than ϵ'.
2) Sort the rectangles of L according to their widths; form m groups of rectangles of approximately equal cumulative heights; round up the widths in each group, to yield a list L_{sup} with $L \leq L_{sup}$.
3) Solve fractional strip packing on L_{sup} with tolerance 1, as in [10].
4) From the fractional strip packing, construct an integral strip packing of L_{sup} and hence a well-structured strip packing of L.
5) Sort L_{narrow} according to decreasing heights and add the rectangles of L_{narrow} to the strip packing of L using the Modified Next Fit Decreasing height Heuristic.

Remark on guillotine cuts : we remark that this algorithm can be performed in 5 stages of guillotine cuts. First, we perform the horizontal cuts which perform the layers.

Secondly, in each layer j, we perform a vertical cut at C_j, thus separating the part reserved to the wide rectangles form the part R_j for the narrow rectangles, and also perform all the vertical cuts defining the columns of configuration used in layer j.

Thirdly, we cut the columns with horizontal cuts, and in R_j, we cut all the sub levels with horizontal cuts.

Fourthly, we finish cutting out the wide rectangles using vertical cuts to adjust the widths of their true values and in each sublevel, we perform vertical cuts corresponding to the narrow rectangles.

Finally, in each sublevel, we perform horizontal cuts to finish cutting out the narrow rectangles. All in all, we have used 5 stages of guillotine cuts.

3 The analysis

The running time is clearly polynomial in n and $1/\epsilon$. Its bottleneck lies in the resolution of the fractional bin-packing problem in step 3, which can be done as in [10] or [13]. Thus the main difficulty in the analysis consists in showing that the strip packing is close to optimal.This is done through a series of lemmas.

Lemma 3 *The list L_{sup} obtained after the grouping of step 2 is such that*

$$lin(L_{sup}) \leq lin(L)(1 + 1/(m\epsilon'))$$

and

$$s(L_{sup}) \leq s(L)(1 + 1/(m\epsilon')),$$

where $s(L)$ is the area of L and $lin(L)$ is the height of the optimal fractional strip packing of L.

Proof: Define the following extension of our partial order on lists of rectangles: $L \leq L'$ if the stack associated to L (used for the grouping), viewed as a region of the plane, is contained in the stack associated to L'. Thus, $L \leq L'$ clearly implies $lin(L) \leq lin(L')$ and $s(L) \leq s(L')$. We now define lists L'_{inf} and L'_{sup} such that $L'_{inf} \leq L \leq L_{sup} \leq L'_{sup}$. These lists are obtained by first cutting the threshold rectangles using the lines $y = ih(L)/m$ $(1 \leq i \leq m-1)$, then considering the m subsequent groups of rectangles in turn (where each group now has cumulative height exactly $h(L)/m$); to define L'_{sup}, we round the widths in each group up to the widest width of the group (up to 1 for the first group); to define L'_{inf}, we round the widths within each group down to the widest width of the next group (down to 0 for the last group). It is easy to see that $L'_{inf} \leq L \leq L_{sup} \leq L'_{sup}$. Moreover, the fractional bin-packing problem for L'_{inf} is almost the same as for L'_{sup} : the stack associated to L'_{sup} is the union of a bottom part of width 1 and height $h(L)/m$, and a top part which is a translated copy of the stack of L'_{inf}. This implies

$$lin(L'_{sup}) = lin(L'_{inf}) + h(L)/m$$

and

$$s(L'_{sup}) = s(L'_{inf}) + h(L)/m.$$

Finally, we note that since all rectangles have width at least ϵ', we have $h(L)\epsilon' \leq s(L) \leq lin(L)$. This implies the statement of the lemma.

$\diamond$

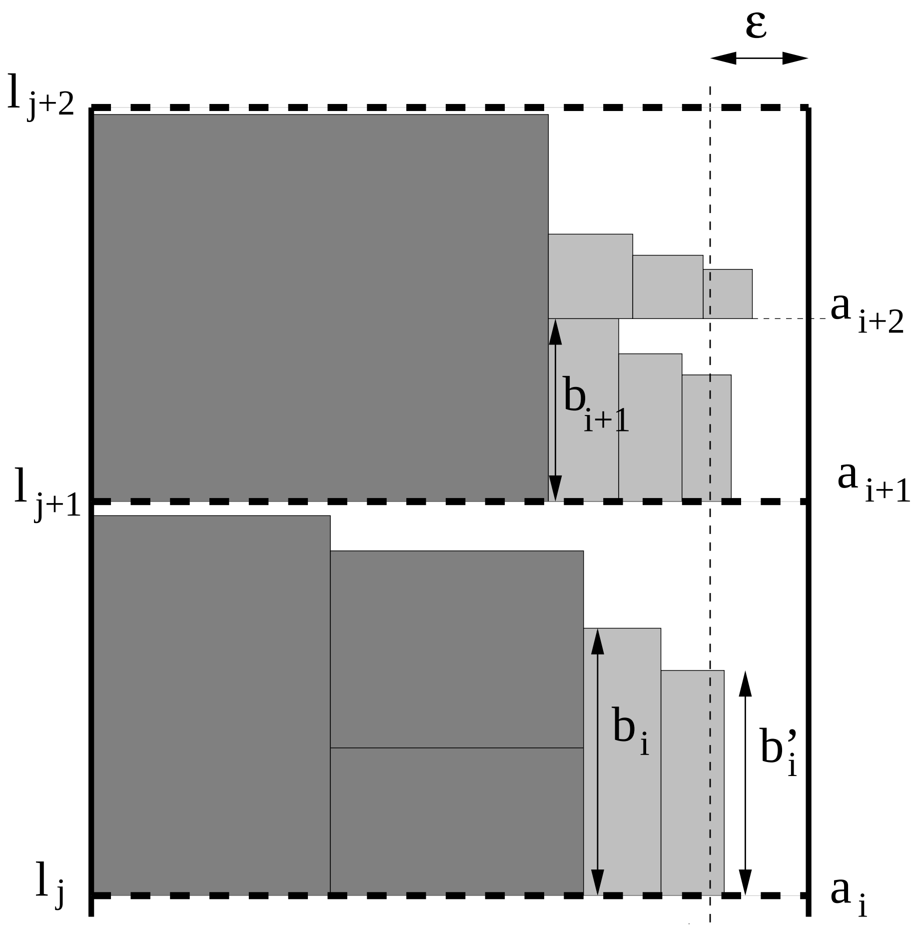

Figure 4: addition of narrow rectangles : notations. Notice that the rectangle $(1-\epsilon') \times b'_i$ is completely covered. Moreover this rectangle is larger than the rectangle $(1-\epsilon') \times b_{i+1}$. This is the main argument of the proof of Lemma 4.

Lemma 4 *Let L_{aux} be the list formed from the union $L_{Sup} \cup L_{narrow}$.*

If the height h_{final} at the end of step 5 is larger than the height h' of the packing of the wide rectangles, then: $h_{final} \leq s(L_{aux})/(1-\epsilon') + 2m + 1$.

To prove this lemma, as in the proof of Theorem 2, we will charge the surface of a sublevel to the rectangles in the sublevel immediately below it.

Proof: Assume that the height h_{final} at the end of step 5 is larger than the height h' of the packing of the wide rectangles.

Let $(a_1, a_2, ..., a_r)$ be the ordered sequence of the lower boundaries of the successive levels constructed by Modified NFDH when inserting the narrow rectangles (hence $a_1 < a_2 < ... < a_r$) and let b_i (respectively b'_i) be the height of the first (respectively last) narrow rectangle placed on the ith level (see figure 4). By definition of NFDH, sublevel i is closed only when the next narrow rectangle is too wide to fit in the current level, which must thus have right over width less than ϵ'. Since rectangle R'_j is completely covered by wide rectangles and all the narrow rectangles placed on sublevel i have height at least b'_i, the total surface occupied on sublevel i is at least $b'_i(1-\epsilon')$¿ Thus the total surface occupied by L_{aux} is at least $\sum_{1\leq i\leq r} b'_i(1-\epsilon')$, which is not smaller than $\sum_{1\leq i\leq r-1} b_{i+1}(1-\epsilon')$, since b_{i+1} is the height of a narrow rectangle considered after the narrow rectangle of height b'_i.

Now, at each interface between layers, there is a height of at most 2 which is not used by the sublevels of NFDH. Thus, the final height is at most $\sum_{1\leq i\leq r} b_i + 2m \leq \sum_{1\leq i\leq r-1} b_i + 2m + 1$. Consequently,

$$s(L_{aux}) \geq (1-\epsilon')(h_{final} - 2m - 1).$$

This last inequality gives the lemma.

$\diamond$

Notice that Lemma 3 obviously induces that $s(L_{auxiliary}) \leq s(L_{general})(1 + 1/m\epsilon')$, which yields the following corollary :

Corollary 1 *With the hypothesis of lemma 4, we have the inequality :*

$$h_{final} \leq s(L_{general})(1 + 1/(m\epsilon'))/(1-\epsilon') + 2m + 1$$

3.1 The overall analysis :

Let h_{final} be the height of the packing of $L_{general}$ constructed by our algorithm. Then

$$h_{final} \leq \max(h', s(L_{general})(1 + 1/(m\epsilon'))/(1-\epsilon') + 2m + 1),$$

where h' is the height of the strip packing constructed in step 4. But $h' \leq h + m$, where h is the height of the fractional strip-packing constructed step 3. By Karp and Karmakar's theorem (with tolerance 1), h is at most $lin(L_{sup}) + 1$. Noticing that $lin(L) \leq Opt(L)$, we obtain:

$$h' \leq h + m$$

$$
\begin{aligned}
&\leq \; lin(L_{sup}) + 1 + m \\
&\leq \; lin(L)(1 + 1/m\epsilon') + 1 + m, \text{ from Lemma 3} \\
&\leq \; Opt(L)(1 + 1/m\epsilon') + 1 + m, \\
&\leq \; Opt(L_{general})(1 + 1/m\epsilon') + 1 + m.
\end{aligned}
$$

Noticing that $s(L_{general}) \leq Opt(L_{general})$, we obtain :

$$h_{final} \leq Opt(L_{general})(1 + 1/(m\epsilon'))/(1 - \epsilon') + 2m + 1$$

Replacing m and ϵ' by their values, we get

$$h_{final} \leq Opt(L_{general})(1 + \epsilon) + 2(2 + \epsilon)^2/(\epsilon^2) + 1$$

hence the theorem.

4 Remarks

In this paper, we proposed a fully polynomial time approximation scheme for strip packing when the rectangle widths and heights are in $(0, 1]$. In many applications, it makes sense to allow rotations of the rectangles (in the case of cutting window panes out of glass or shapes out of leather, for example). We conjecture that our approach can be extended to solve the strip packing problem when rotations of 90 degrees are allowed. It should however be noted that sometimes the optimal packing may use rotations of angles other than 90 degrees, even in the simple situation when one wants to pack squares in a strip [5], [6].

Finally, since we find the solution developed here relatively simple, we hope that it may help for attacking the three-dimensional version of the problem, as well as other variants of multi-dimensional cutting-stock problems.

Acknowledgments. The authors would like to thank Neal Young for a fruitful discussion which simplified stage 3 of the algorithm. This simplification later enabled us to appeal to the fractional packing results from [11], thus easily rendering the scheme *fully* polynomial.

References

[1] Baker B.S., Brown D.J. and Katseff H.P., A 5/4 algorithm for two-dimensional packing, J. of Algorithms, 2(1981), 348-368.

[2] Baker B.S., Coffman E.G. and Rivest R.L., Orthogonal packings in two dimensions, SIAM J. Comput. 9(1980), 846-855.

[3] Coffman E.G. Jr, Garey M.R. and Johnson, D.S., Approximation Algorithms for Bin-Packing - An Updated Survey, in Algorithm design for computer System design, G. Ausiello, M. Lucertini and P. Serafini editors (1984) 49-106.

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

[5] El Moumni Said, Rangements optimaux de carrés unite dans une bande infinie, Preprint, université de Mons-Hainaut, Institut de Mathématique et d'Informatique, 15, Avenue Maistriau, 7000 Mons, Belgique.

[6] P. Erdos, Graham, On packing squares with equal squares, Journal of Combinatorial Theory A 19(1975), 119-123.

[7] Fernandez de la Vega W., and Lueker G.S., Bin packing can be solved within 1+e in linear time, Combinatorica 1(1981), 349-355.

[8] Fernandez de la Vega W., and Zissimopoulos V., An approximation scheme for strip-packing of rectangles with bounded dimensions, preprint (1991).

[9] P.C. Gilmore and R.E. Gomory, Multistage Cutting Stock Problems of Two and More Dimensions, Operations Research 13 (1965). 94-120.

[10] Golan I., Performance bounds for orthogonal, oriented two-dimensional packing algorithms, SIAM J. Comput. 10(1981), 571-582.

[11] Karmarkar N. and Karp R.M., An Efficient Approximation Scheme for the One-Dimensional Bin-Packing Problem, FOCS (1982), 312-320.

[12] C. Kenyon, E. Remila, Approximate strip packing, FOCS (1996), 30-37.

[13] Plotkin S.A., Shmoys D.B. and Tardos E., Fast Approximation Algorithms for Fractional Packing and Covering Problems, Math. of Op. Research 20, 2, (1995) 257-403

[14] Schiermeyer I., Reverse fit : a 2-optimal algorithm for packing rectangles, Proceedings of the 2^{nd} European Symposium on Algorithms (ESA) (1994), 290-299

[15] Sleator D. D., A 2.5 times optimal algorithm for bin packing in two dimensions, IPL 10(1980), 37-40.

[16] Steinberg A., A strip-packing algorithm with absolute performance bound 2, SIAM J. Comput. 26,2, (1997) 401-409.

[17] Neal E. Young, Randomized Rounding Without Solving the Linear Program, Proceedings of the Symposium on Discrete Algorithms (SODA) (1995), 170-178.