Approximation Algorithms for Multiple Strip Packing[⋆]

Marin Bougeret[1], Pierre Francois Dutot[1], Klaus Jansen[2], Christina Otte[2], and Denis Trystram[1]

[1] LIG, Grenoble University, France
{bougeret,dutot,trystram}@imag.fr
[2] Department of Computer Science
Christian-Albrechts-University Kiel
Christian-Albrechts-Platz 4, 24098 Kiel, Germany.
{kj,cot}@informatik.uni-kiel.de

Abstract. In this paper we study the Multiple Strip Packing (MSP) problem, a generalization of the well-known Strip Packing problem. For a given set of rectangles, $r_1, \ldots, r_n$, with heights and widths ≤ 1, the goal is to find a non-overlapping orthogonal packing without rotations into $k \in \mathbb{N}$ strips $[0,1] \times [0,\infty)$, minimizing the maximum of the heights. We present an approximation algorithm with absolute ratio 2, which is the best possible, unless $\mathcal{P} = \mathcal{NP}$, and an improvement of the previous best result with ratio $2+\varepsilon$. Furthermore we present simple shelf-based algorithms with short running-time and an AFPTAS for MSP. Since MSP is strongly $\mathcal{NP}$-hard, an FPTAS is ruled out and an AFPTAS is also the best possible result in the sense of approximation theory.

Key words: Strip Packing, Scheduling in grids

1 Introduction

In this paper we study the Multiple Strip Packing (MSP) problem, a generalization of the well-known Strip Packing (SP) problem. For a given set of rectangles, $r_1, \ldots, r_n$, with heights and widths ≤ 1, the goal is to find a non-overlapping orthogonal packing without rotations into $k \in \mathbb{N}$ strips $[0,1] \times [0,\infty)$, minimizing the maximum of the heights. As much as Strip Packing, its generalization Multiple Strip Packing is not only of theoretical interest, but also has many applications to real-world problems as in computer grids, server consolidation and in cutting problems. In computer grids for example, MSP is related to the problem of finding a schedule for parallel tasks into different clusters of processors with minimum makespan [12]. Consider an instance $L = \{r_1, \ldots, r_n\}$ of MSP. The value k always denotes the number of strips $S_1, \ldots, S_k$. For $i \in \{1, \ldots, k\}$ the

[⋆] Work supported by EU project "‘AEOLUS: Algorithmic Principles for Building Efficient Overlay Computers"’, EU contract number 015964, and DFG project JA612/12-1, "‘Design and analysis of approximation algorithms for two- and threedimensional packing problems"’, and DGA-CNRS

value h_i denotes the height of a feasible packing in strip S_i. For an algorithm A for MSP let $A(L)$ be the output of the algorithm, in this case the maximum height of the packing generated, i.e. $\max_{i\in\{1,\ldots,k\}} h_i$. The optimal value is denoted with $OPT(L)$, in this case the minimal height that can be achieved. The quality of an approximation algorithm is measured by its performance ratio. For a minimization problem as MSP we say that A has *absolute ratio* α, if $\sup_L {}^{A(L)}/_{OPT(L)} \leq \alpha$, and *asymptotic ratio* α, if $\alpha \geq \limsup_{OPT(L)\to\infty} {}^{A(L)}/_{OPT(L)}$, respectively. A minimization problem admits an *(asymptotic) polynomial-time approximation scheme* ((A)PTAS), if there exists a family of polynomial-time approximation algorithms $\{A_\varepsilon | \varepsilon > 0\}$ of (asymptotic) $(1+\varepsilon)$-approximations. We call an approximation scheme *fully polynomial* ((A)FPTAS), if the running-time of every algorithm A_ε is bounded by a polynomial in n and $\frac{1}{\varepsilon}$. Zhuk showed in [16] that there is no approximation algorithm for MSP with absolute ratio less than 2. Since MSP can be reduced to 3-Partition, it is also strongly $\mathcal{NP}$-hard. Therefore a PTAS and an FPTAS are ruled out and an AFPTAS is asymptotically the best possible.

A related problem is 3D Strip Packing (3SP), which also is a generalization of Strip Packing. Here the goal is to find a packing of a given list of cuboids with side lengths bounded by one into a 3-dimensional strip $[0,1] \times [0,1] \times [0,\infty)$, minmizing the height of the packing. Multiple Strip Packing with k strips can be reduced to 3SP by introducing a cuboid with depth $1/k$ for each rectangle packing the strips next to each other.

Parallel Job Scheduling in Grids with identical machines is also a related problem. In the offline case we have m machines M_i with ℓ processors and jobs $j \in J$ with processing time p_j, and a size $size_j$. The jobs must be executed on parallel processors within one machine M_i, but not necessary on consecutive processors. The machines can be seen as strips with width l and the jobs as vertically scissile rectangles with width $size_j$ and height p_j. In Multiple Strip Packing we have just the additional constraint that a job must be scheduled on consecutive processors. Unfortunately this is the reason why approximation algorithms for Parallel Job Scheduling cannot be applied to MSP maintaining their ratio.

Known Results. Multiple Strip Packing was first considered by Zhuk [16], who showed that there is no approximation algorithm with abolute ratio better than 2, and later by Ye et. al. [15]. Both concentrated on the online case. Additonally an approximation algorithm for the offline case with ratio $2+\varepsilon$ was achieved in [15]. For Strip Packing Coffman et al. gave in [8] an overview about performance bounds for shelf-orientated algorithms as $NFDH$ (Next Fit Decreasing Height) and $FFDH$ (First Fit Decreasing Height). Those adopt an absolute ratio of 3, and 2.7, respectively. Schiermeyer [11] and Steinberg [13] presented independently an algorithm for SP with absolute ratio 2. A further important result is an AFPTAS for SP with additive constant $\mathcal{O}(1/\varepsilon^2)$ of Kenyon and Rémila [9]. This constant was improved by Jansen and Solis-Oba, who presented in [7] an APTAS with additive constant 1. For 3SP Jansen and Solis-Oba obtained an algorithm with ratio $2+\varepsilon$ in [6] as an improvement of the formerly known result by Miyazawa and Wakabayashi [10], who presented an algorithm with asymptotic ratio at most 2.64. Bansal et al. presented in [2] an algorithm for $3SP$ with a ratio of $T_\infty \approx 1.69$, which is the best known result. Schwiegelshohn et al. [12] achieved ratio 3 for a version of Parallel Job Scheduling in Grids without release times, and ratio 5 with release times. Tchernykh et al. presented in [14] an algorithm with absolute ratio 10 for the case of machines with different numbers of processors and without release times. However, this algorithm cannot be applied directly to MSP because of the non-contiguity.

Our Results. In this paper we present an approximation algorithm with absolute ratio 2, which is an improvement of the former result of $2+\varepsilon$ by Ye et al. [15] and best possible, unless $\mathcal{P} = \mathcal{NP}$. We also introduce an AFPTAS for Mutiple Strip Packing, which is a generalization of the algorithm of Kenyon and Rémila [9]. Our algorithm achieves an additive constant of $\mathcal{O}(1)$, if the number of strips is sufficient large, otherwise an additive constant of $\mathcal{O}(1/\varepsilon^2)$. Furthermore we show how to use the simple shelf-based heuristics $NFDH$ and $FFDH$ to obtain approximation algorithms for MSP with the same asymptotic ratio as for SP.

Organisation of the Paper. In the next section we introduce two shelf-based algorithms, using Next Fit and First Fit policies. In Section 3 we present a 2-approximation for MSP. Here we distinguish between different sizes for k. For $k = 1$ we use the 2-approximation of Steinberg [13] or Schiermeyer [11]. If $k = 2$ or bounded by a specified constant c we make use of a result by Bansal et al. [1] for Rectangle Packing with Area Maximization (RPA). For $k \geq c$ we use an approximation algorithm for 2D bin packing with asymptotic ratio 1.69 of Caprara [3]. In the last section we present an AFPTAS for MSP. Here we generalize the algorithm by Kenyon and Rémila [9]. Interestingly, the additive constant in our AFPTAS can be reduced from $\mathcal{O}(1/\varepsilon^2)$ to $\mathcal{O}(1)$, if the number k of strips is large enough.

2 Shelf-based algorithms

In this section we modify the shelf-based heuristics NFDH and FFDH. [8]. A shelf is a row of items placed next to each other left-justified. The baseline of a shelf is either the bottom of the bin or the extended upper edge of the tallest item packed in the shelf below. NFDH generates for a given list of rectangles $L = \{r_1, \dots, r_n\}$ a packing into a strip with height at most $2OPT_{SP}(L)+h_{\max}$, FFDH produces a packing of height at most $1.7OPT_{SP}(L)+h_{\max}$, where $OPT_{SP}(L)$ is the optimum value of Strip Packing for the instance L and $h_{\max}$ is the height of the tallest item in L. Via this modification we obtain approximation algorithms for Multiple Strip Packing with the same asymptotic ratios. Furthermore, we present another algorithm, that computes for rectangles with widths bounded by $\varepsilon < 1$ a packing of height at most $1/(1-\varepsilon) OPT(L) + 2h_{\max}$.

Theorem 1. *Let A be one of the shelf-based Strip Packing algorithms NFDH or FFDH with asymptotic ratio $\alpha > 1$, that creates for an instance L a packing of height less than $\alpha OPT_{SP}(L) + h_{\max}$. For any $k \in \mathbb{N}$ there exists an algorithm A_k that packs a list of rectangles L into k strips with $A_k(L) \leq \alpha OPT(L) + h_{\max}$.*

Proof. For any instance L of MSP we define the algorithm A_k as follows

1 Pack the sorted rectangles with A into one strip S. (In particular the rectangles are first sorted by non-increasing height.) Let $A(L)$ denote the height of S.
2 Cut out the first shelf and pack it into the first strip S_1.
3 Divide the residual strip S into k parts:
    3.1 For each $\ell \in \{0, 1, \ldots, k\}$ draw a horizontal line across S at height $\ell(A(L) - h_{\max})/k$.
    3.2 For $\ell \in \{0, 1, \ldots, k-1\}$ pack all items intersecting the ℓth line and all items between the ℓth and $(\ell+1)$th line into strip $S_{\ell+1}$.

We show now that for any instance L of MSP the output of A_k is less than $\alpha\mathrm{OPT}(L) + h_{\max}$. Let $t \in \mathbb{N}$ be the number of shelves produced by A in Step 1 and H_j, $j \in \{1, \ldots, t\}$, the height of the jth shelf. Since there are no items intersecting the 0th line (see Fig 1), the height h_1 of the first strip S_1 is bounded by $h_{\max} + \frac{1}{k}\left(\sum_{j=1}^{t} H_j - h_{\max}\right)$ after the last step of the algorithm. For a strip S_i, $i \in \{2, \ldots, k\}$, containing the items between the $(i-1)$th and the ith line and the ones intersecting the $(i-1)$th line, we have $h_i \leq \frac{\sum_{j=1}^{t} H_j - h_{\max}}{k} + h_{\max} = \frac{A(L) - h_{\max}}{k} + h_{\max}$. We conclude

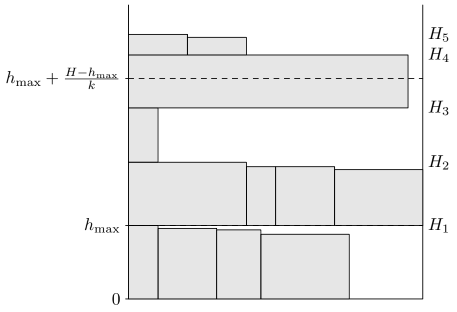

Fig. 1. Dividing strip S.

$$A_k(L) = \max_{i \in \{1, \ldots, k\}} h_i \leq \frac{A(L) - h_{\max}}{k} + h_{\max}$$
$$\leq \frac{\alpha\mathrm{OPT}_{SP}(L) + h_{\max} - h_{\max}}{k} + h_{\max} = \frac{\alpha\mathrm{OPT}_{SP}(L)}{k} + h_{\max}.$$

Since ${}^{1}\!/_{k}\mathrm{OPT}_{SP}(L)$ is a lower bound for $\mathrm{OPT}(L)$ the proof is complete.

The running-time of the above algorithm is $\mathcal{O}(n \log n)$.

Corollary 1. *Let L be an instance of MSP. In a packing generated by the above algorithm A_k we have $\max_{i\in\{1,\dots,k\}} |h_i - A_k(L)| \le 2h_{\max}$, where h_i denotes the height of strip S_i.*

Another way to pack a set of rectangles with a modified version of the $NFDH$ heuristic into k strips is the following:

Algorithm 2

```
1 Sort the rectangles by non-increasing height.
2 For each $i \in \{1,\dots,k\}$ pack one shelf according to the $NFDH$ heuristic into
  strip $S_i$, that means starting in the lower left corner pack the rectangles next
  to each other on the baseline of strip $S_i$, until the next rectangle does not fit.
  Draw a new baseline at the top edge of the tallest rectangle (that clearly is
  the first one).
3 Take the strip $S^-$ with the current lowest height $h^-$ and pack one shelf
  according to the $NFDH$ heuristic on top of the shelves.
4 Repeat Step 3 until all rectangles are packed.
```

The packing generated by the above algorithm is very smooth, in the sense that the heights of the strips only differ by $h_{\max}$.

Lemma 1. *For a set of rectangles $L = \{r_1, \dots, r_n\}$ Algorithm 2 with output $A(L)$ generates a packing into k strips, so that $\max_{i\in\{1,\dots,k\}} |A(L) - h_i| \le h_{\max}$.*

This leads to a further result about rectangles with bounded width. Coffman et al. showed in [8] that $FFDH$ applied to an instance L of rectangles with widths bounded by $1/m$ for some integer m generates a packing into a strip of height at most $(1 + \frac{1}{m})OPT_{SP}(L) + h_{\max}$. Our result for packing into k strips is the following:

Theorem 3. *For a set of rectangles $L = \{r_1, \dots, r_n\}$ with widths bounded by $\varepsilon > 0$ we obtain by the Algorithm 2 with output $A(L)$ a packing into k strips with height less than $\frac{1}{1-\varepsilon}OPT(L) + 2h_{\max}$.*

For $\varepsilon = \frac{1}{m}$ this is equal to $A(L) \le \left(1 + \frac{1}{m-1}\right)\mathrm{OPT}(L) + 2h_{max}$.

3 A two-approximation for MSP

In this section we construct a polynomial-time approximation algorithm for MSP with absolute ratio 2. Since there is no approximation algorithm for MSP with ratio smaller than 2 (unless P=NP), this is the best possible result. To handle different sizes of k we use, besides the well-known algorithms of Steinberg [13] or Schiermeyer [11], a result of Bansal et al. [1] for Rectangle Packing with Area Maximization (RPA) and a of Caprara [3].

3.1 One or two strips

The case $k = 1$ is trivial, because we can use the algorithm of Steinberg [13] or Schiermeyer [11] with absolute performance bound 2.

Theorem 4 (Steinberg [13]). *Let $L = \{r_1, \dots, r_n\}$ be a set of rectangles with heights h_i and widths w_i and Q be a rectangle with width u and height v. Let $h := \max_{i \in \{1,\dots,n\}} h_i$ and $w := \max_{i \in \{1,\dots,k\}} w_i$. If the following inequalities hold,*

$$w \leq u, \quad h \leq v, \quad 2SIZE(L) \leq uv - (2w - u)_+(2h - v)_+ \tag{1}$$

then it is possible to pack L into the rectangle Q.(As usual, $x_+ = \max(x, 0)$.)

Therefore let us first consider the case for $k = 2$. Here we use the PTAS found by Bansal et al. [1] for RPA. In RPA we are given a set of rectangles $L = \{r_1, \dots, r_n\}$ with widths w_i and heights h_i and a bin of unit size. The goal is to find a feasible packing of a subset $L' \subset L$ of the rectangles and to maximize the area of the rectangles in L'.

Algorithm 5

1 *Guess the height of an optimal solution for MSP and denote it with v.*
2 *Scale the heights of the rectangles in L by $1/v$ so that the corresponding packing fits into one bin of height and width one.*
3 *The set of resulting rectangles L_v is now considered as an instance of RPA with $OPT_{RPA}(L) = SIZE(L_v)$, where $SIZE(L_v)$ is the total area of all rectangles in L_v. Apply the algorithm in [1] with accuracy $\varepsilon = 1/2$ and find a packing of a subset $L'_v \subset L_v$ with total area at least $(1 - \varepsilon)SIZE(L_v)$. By rescaling the rectangles of L'_v get a packing for the first strip with height at most v.*
4 *Since $SIZE(L_v) \leq 2$ the remaining items in $L_v \backslash L'_v$ have total area $SIZE(L_v \backslash L'_v) \leq \varepsilon SIZE(L_v) \leq 1$. Therefore we can pack them with Steinberg's algorithm into a strip of height at most 2. Rescaling gives us a second strip of height at most $2v$.*

The running-time of the algorithm is polynomial in n:
In the first step we can assume that the heights of the rectangles are rational, so by multiplying with a common denominator they become integer values. Then the optimum height v of MSP is also integer and equals a sum of heights of the rectangles in L, so we have $h_{\max} \leq v \leq nh_{\max}$. Thus Binary Search takes at most $\log(nh_{\max})$ iterations to find the value v. Step 3 is also polynomial, since we apply the algorithm in [1] for a fixed accuracy $\varepsilon = 1/2$.

3.2 A bounded number of strips

In the case of a constant number of strips we can use an extended version of the PTAS for RPA in [1] called $kRPA$. Another helpful tool is the next lemma. The proof can be obtained applying Steinberg's algorithm for $h, w, u = 1$ and $v = k/2$ in equation 1.

Lemma 2. *If L is an instance of 2DBP with total area $SIZE(L) \leq {}^{k}/_{4}$ and $k \geq 3$, then there exists a packing of L into k bins.*

Algorithm 6

1 *Guess an optimal height for MSP and denote it with v.*

2 *Scale the heights of the rectangles in L by ${}^{1}/_{v}$ so that the corresponding packing fits into k bins of height and width one.*

3 *The set of resulting rectangles L_v is now considered as an instance of RPA with $OPT_{RPA}(L) = SIZE(L_v)$. Apply kRPA to k bins of unit size and find for an accuracy $\varepsilon \leq {}^{1}/_{4}$ a packing for a subset $L'_v \subset L_v$ with total area $(1-\varepsilon)SIZE(L_v)$. By rescaling the rectangles of L'_v we get k bins of height v.*

4 *For the total area of the remaining rectangles in $L_v \backslash L'_v$ we have $SIZE(L_v \backslash L'_v) = \varepsilon SIZE(L_v) \leq {}^{k}/_{4}$. Pack those rectangles according to Lemma 2 into k bins and rescale the rectangles. This results again in k bins of height at most v.*

5 *Stack every two bins on top of each other and get a solution with k bins of height at most $2v$.*

3.3 A large number of strips

Caprara presented in [3] a shelf algorithm for $2DBP$ that produces a solution whose asymptotic ratio can be made arbitrarily close to $T_\infty = 1.69...$. Clearly if the number of strips is large enough ($\approx 10^4$) applying this algorithm we get a two-approximation for MSP stacking every two bins on each other. Alternatively, we can use the recently published two-approximation for 2DBP by Jansen et al. [5] to achieve this result. Along with the previous sections we have the following:

Theorem 7. *For any $k \in \mathbb{N}$ there is a polynomial-time algorithm for MSP with absolute ratio two.*

4 An AFPTAS for MSP

In this section we present an AFPTAS for MSP. The algorithm is a generalization of an AFPTAS found by Kenyon and Rémila [9] for Strip Packing. For an instance L of Strip Packing and an accuracy $\varepsilon > 0$ their algorithm generates a packing with height $(1+\varepsilon)OPT_{SP}(L) + \mathcal{O}({}^{1}/_{\varepsilon^2})h_{\max}$. Our algorithm achieves the same ratio for Multiple Strip Packing. For instances with k sufficient large, namely $k \in \Omega({}^{1}/_{\varepsilon^3})$, our algorithm adopts an improved additive constant of $\mathcal{O}(1)$. More precisely for an accuracy ε and $k \geq \lceil {}^{128}/_{\varepsilon^3} \rceil$ we get an approximation ratio of $(1+\varepsilon)\mathrm{OPT}(L) + 6h_{\max}$.

4.1 The regular case

As in Section 2 we divide a packing into one strip into k parts of nearly the same height and distribute them to k strips.

Theorem 8 (Kenyon & Rémila [9]). *For a list $L = \{r_1, \ldots, r_n\}$ of rectangles and an accuracy $\varepsilon > 0$ the algorithm A_ε^{KR} in [9] generates a packing into one strip with height at most $(1+\varepsilon)OPT_{SP}(L) + (4(\frac{2+\varepsilon}{\varepsilon})^2 + 1)h_{\max}$.*

Our result is the following:

Theorem 9. *For a list $L = \{r_1, \ldots, r_n\}$ of rectangles with widths and heights ≤ 1 and an accuracy $\varepsilon > 0$ there exits an algorithm A_ε that generates a packing into k strips, so that $A_\varepsilon(L) \leq (1+\varepsilon)OPT(L) + (2(\frac{2+\varepsilon}{\varepsilon})^2 + 2)h_{\max}$.*

4.2 Instances with a large number of strips

In this section we consider the case $k \geq \lceil {}^{128}/_{\varepsilon^3} \rceil$. In this case it is possible to improve the additive constant to $\mathcal{O}(1)h_{\max}$ by balancing the configurations.

Rounding. We choose $\varepsilon' = {}^{\varepsilon}/_{4}$ (w.l.o.g. ${}^{1}/_{\varepsilon'}$ integral) and divide the list of rectangles L into a list of narrow rectangles $L_{narrow} := \{r_i \in L | w(r_i) \leq \varepsilon'\}$ and a list of wide rectangles $L_{wide} := \{r_i \in L | w(r_i) > \varepsilon'\}$. Then we round L_{wide} to an instance L_{sup} with only $M := ({}^{1}/_{\varepsilon'})^2$ different widths. For the rounding step we put the wide rectangles sorted by non-increasing widths left-aligned on a stack. Let $STACK(L)$ denote the total area of the plane covered by this stack and let H denote its height. Moreover, for arbitrary lists L'', L' we define a relation $\leq_g$, so that $L'' \leq_g L'$, if and only if $STACK(L'') \subseteq STACK(L')$. We draw $M-1$ horizontal lines through $STACK(L)$ with distance ${}^{H}/_{M}$ starting at the bottom. Therefore we get M so-called *threshold* rectangles. A rectangle is a threshold rectangle if it either with its interior or with its lower edge intersects a line at height ${}^{iH}/_{M}$, $i \in \{1, \ldots, M-1\}$. For $i \in \{1, \ldots, M-1\}$ we round up the width of each rectangle between the lines ${}^{iH}/_{M}$ and ${}^{(i+1)H}/_{M}$ to the width of the ith threshold rectangle. The widths of the rectangles below the first line are rounded up to the width of the undermost rectangle in the stack. So we get at most M groups of different widths (see Fig 2). Furthermore, we get a list L_{sup} of rectangles with widths larger than ε' and only M different widths, in particular we have $L_{wide} \leq_g L_{sup}$.

Fractional Packing. Our first objective is to create a fractional packing for the wide rectangles into k strips. To do this we introduce *configurations.* A configuration is a non-empty multiset of widths, which sum up to less than one. Denote with q the number of different configurations C_j with height x_j. Let α_{ij} be the number of occurence of width w_i in configuration C_j and let β_i be the

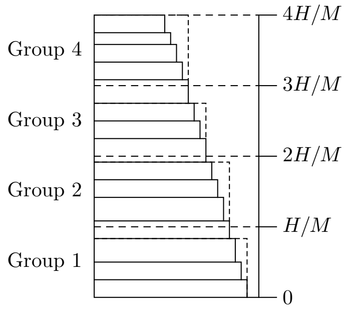

Fig. 2. Rounding the rectangles in L_{sup}.

total height of all rectangles of width w_i. Based on the solution of the following Linear Program

$$\begin{aligned} &\min \frac{\sum_{j=1}^{q} x_j}{k} \\ \text{s.t.} &\sum_{j=1}^{q} \alpha_{ij} x_j \geq \beta_i \text{ for all } i \in \{1, \dots, M\} \\ &x_j \geq 0 \text{ for all } j \in \{1, \dots, q\}, \end{aligned} \tag{LP(L_{sup})}$$

by distributing the configurations to k strips we get the requested fractional packing for the rectangles in L_{sup}. Note that $\operatorname{rank}(\alpha_{ij})_{ij} \leq M$ and hence a basic solution x of $LP(L_{sup})$ has at most M nonzero entries. In the next section we show how to transform a fractional packing into a feasible packing for L_{sup}. Later the rectangles in L_{narrow} are packed into the idle space in a Greedy manner. For a list L of rectangles let $LIN(L)$ denote the height of an optimum fractional packing for L. Let $h_0 := LIN(L_{sup})$ and note that $h_0 \leq OPT(L)$.

Lemma 3. *Let $x = (x_1, \dots, x_q)$ be a solution of $LP(L_{sup})$ with at most $m \leq M$ nonzero entries $x_1, \dots, x_m$. For $k \geq \lceil 128/\varepsilon^3 \rceil$ we get a fractional packing into k strips with height at most $(1 + \varepsilon')h_0$ and at most $m' \leq 2M$ different configurations.*

Proof. First we fractionally pack the rectangles into the configurations. Imagine each configuration C_j as a bin with height x_j and width c_j and divide it into α_{ij} columns of widths w_i and height x_j. Pack the rectangles in L_{sup} of width w_i in a Greedy manner fractionally into the columns of width w_i until exactly height x_j, starting with $j = 1$. In this way each column contains a sequence of rectangles, which completely fits inside the column, and possibly the top part of a rectangle, that started in a previous column, and the bottom part of a rectangle, that is too tall to fit into this column. Since $\sum_{j=1}^{m} \alpha_{ij} x_j \geq \beta_i$, there will be maybe more than enough space for the rectangles of width w_i in the configurations. In this case we distribute the rectangles among the columns and delete the additional space. So we split a configuration C_j into two parts, one of the old type where the columns of width w_i are completely filled and one without columns of width w_i. This case may happen only M times. So we have in total $m' = m + M \leq 2M$ configurations $C_1, \dots C_{m'}$ with nonzero heights $x_1, \dots, x_{m'}$.

Notice that there exist configurations with height larger or equal h_0, since if not we conclude $\sum_{j=1}^{m'} x_j < m'h_0 \leq 2Mh_0 \overset{\varepsilon'=\varepsilon/4}{=} \frac{32h_0}{\varepsilon^2} < kh_0$, which is a contradiction. Consider a configuration C_j, $j \in \{1, \dots, m'\}$. If $x_j \geq h_0$ we allocate $\lfloor x_j/h_0 \rfloor$ empty strips with height h_0 for C_j. If then $x_j/h_0 - \lfloor x_j/h_0 \rfloor \leq \varepsilon' h_0$, we assign to C_j additional space with height $(x_j/h_0 - \lfloor x_j/h_0 \rfloor)$ in a strip, that has already height h_0. If $x_j/h_0 - \lfloor x_j/h_0 \rfloor > \varepsilon' h_0$, we divide $(x_j/h_0 - \lfloor x_j/h_0 \rfloor)$ into at most $1/\varepsilon'$ stripes with height less or equal $\varepsilon' h_0$. So assign to C_j additional space of height $\varepsilon' h_0$ in no more than $1/\varepsilon'$ strips, which are already occupied until height h_0. In the same way as the remaining stripes we handle configurations of height less than h_0. Since there are at most $2M$ configurations with nonzero height, we get at most $2M/\varepsilon' = 2/\varepsilon'^3 \leq 2\cdot4^3/\varepsilon^3 \leq k$ additional assignments of height $\varepsilon' h_0$, which can be distributed to k strips. Thus by this assignment policy, where the configurations are balanced, each strip has allocated area of height at most $(1+\varepsilon')h_0$ for at most 2 different configurations.

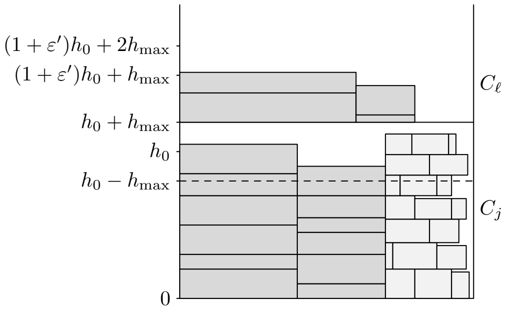
$(1+\varepsilon')h_0 + 2h_{\max}$
$(1+\varepsilon')h_0 + h_{\max}$
$h_0 + h_{\max}$
h_0
$h_0 - h_{\max}$
0
C_ℓ
C_j

Fig. 3. S_i with C_j and C_ℓ.

Integral Packing. The next Lemma shows how to get from a fractional packing to a feasible integral packing. A proof is given in the full paper.

Lemma 4. *Let $x = (x_1, \dots, x_q)$ be a solution of $LP(L_{sup})$ with at most $m' \leq 2M$ nonzero entries $x_1, \dots, x_{m'}$. For $k \geq \lceil 128/\varepsilon^3 \rceil$ we can convert x to a feasible packing for the wide rectangles with height at most $(1+\varepsilon')h_0 + 2h_{\max}$ and at most 2 different configurations per strip.*

Since we can guarantee that there are at most 2 different configurations per strip, the additive constant will be improved, while the running-time is still polynomial in n and $1/\varepsilon$.

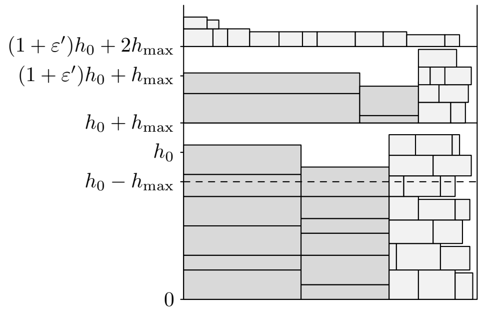
$(1+\varepsilon')h_0 + 2h_{\max}$
$(1+\varepsilon')h_0 + h_{\max}$
$h_0 + h_{\max}$
h_0
$h_0 - h_{\max}$
0

Fig. 4. S_i after packing the narrow rectangles.

Our last step is to pack the narrow rectangles. We use a modified version of the NFDH algorithm: For strip S_i as above we pack narrow rectangles with NFDH into the empty space next to the configurations until the total height is at most $(1+\varepsilon')h_0+2h_{\max}$. After that we repeat the process for strip S_{i+1}. When all strips are filled in this way, we draw a horizontal line at height $(1+\varepsilon')h_0+2h_{\max}$ in each strip and pack the remaining narrow rectangles with Algorithm 2 on top (see Fig 3 and 4). Thus we can ensure by Lemma 1 that the maximum difference of the heights of two arbitrary strips is at most $h_{\max}$ (see Fig 4). Let h_{final} denote the height of the packing after packing the narrow rectangles.

Lemma 5. *Let $k \geq \lceil 128/\varepsilon^3 \rceil$. If $h_{final} \geq (1+\varepsilon')h_0 + 2h_{\max}$, then we have $h_{final} \leq \frac{SIZE(L_{sup} \cup L_{narrow})}{k(1-\varepsilon')} + 6h_{\max} + \varepsilon' h_0$.*

For details we refer to the full paper. The next lemma is shown in [9] for the Linear Program corresponding to Strip Packing, but obviously also holds for our linear program $LP(L_{sup})$.

Lemma 6. *[9] For the rounded instance L_{sup} and L_{wide} the inequalities $LIN(L_{sup}) \leq LIN(L_{wide})\left(1+\frac{1}{M\varepsilon'}\right)$ and $SIZE(L_{sup}) \leq SIZE(L_{wide})\left(1+\frac{1}{M\varepsilon'}\right)$ hold.*

The entire algorithm is now defined as follows:

Algorithm 10

```
1 Set $\varepsilon' := \varepsilon/4$ and $M := (1/\varepsilon')^2$.
2 Partition $L$ into $L_{wide}$ and $L_{narrow}$.
3 Construct $L_{sup}$, so that $L_{wide} \leq_g L_{sup}$ and there are only $M$ different widths
  in $L_{sup}$.
4 Solve the linear program $LP(L)$.
5 Construct a feasible solution for $L_{sup}$ by balancing the configurations.
6 Use modified $NFDH$ to pack the rectangles in $L_{narrow}$ into the remaining
  space and on top of the strips.
```

Theorem 11. *If $k \geq \lceil 128/\varepsilon^3 \rceil$ the Algorithm 10 generates for an instance L of MSP a packing of height at most $(1+\varepsilon)OPT(L) + \mathcal{O}(1)h_{\max}$.*

References

1. N. Bansal, A. Caprara, K. Jansen, L. Prädel, and M. Sviridenko. How to maximize the total area of rectangle packed into a rectangle. To appear in *The 20th International Symposium on Algorithms and Computation (ISAAC 2009)*, 2009.
2. N. Bansal, X. Han, K. Iwama, M. Sviridenko, and G. Zhang. Harmonic algorithm for 3-dimensional strip packing problem. In *Proceedings of the eighteenth ACM-SIAM symposium on Discrete algorithm (SODA 2007)*, pages 1197–1206, 2007.
3. A. Caprara. Packing 2-dimensional bins in harmony. In *Proceedings of the 43 rd IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pages 490–499, 2002.
4. A. Caprara, A. Lodi, and M. Monaci. An approximation scheme for the two-stage, two-dimensional bin packing problem. In *Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 315–328, 2002.
5. K. Jansen, L. Prädel, and U. M. Schwarz. Two for one: Tight approximation of 2d bin packing. In *Algorithms and Data Structures Symposium (WADS 2009)*, 2009.
6. K. Jansen and R. Solis-Oba. An asymptotic approximation algorithm for 3d-strip packing. In *Proceedings of the seventeenth ACM-SIAM symposium on Discrete algorithm (SODA 2006)*, pages 143–152, 2006.
7. K. Jansen and R. Solis-Oba. New approximability results for 2-dimensional packing problems. In *Journal of Discrete Optimization*, 6:310-323, 2009.
8. E. G. Coffman Jr., M. R. Garey, D. S. Johnson, and R. E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal of Computing*, 9(4):808–826, 1980.
9. C. Kenyon and E. Rémila. A near optimal solution to a two-dimensional cutting stock problem. *Mathematics of Operations Research*, 25:645–656, 2000.
10. F. Miyazawa and Y. Wakabayashi. Cube packing. In *Latin Theoretical Informatics 2000 (LATIN 2000)*, pages 58–67, 2000.
11. I. Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *Proceedings of the Second European Symposium on Algorithms*, pages 290–299. Springer-Verlag, 1994.
12. U. Schwiegelshohn, A. Tchernykh, and R. Yahyapour. Online scheduling in grids. In *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1–10, 2008.
13. A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal of Computing*, 26(2):401–409, 1997.
14. A. Tchernykh, J. Ramírez, A. Avetisyan, N. Kuzjurin, D. Grushin, and S. Zhuk. Two level job-scheduling strategies for a computational grid. In *PPAM*, pages 774–781, 2005.
15. D. Ye, X. Han, and G. Zhang. On-line multiple-strip packing. In *The 3rd Annual International Conference on Combinatorial Optimization and Applications(COCOA 2009)*, 2009.
16. S.N. Zhuk. Approximate algorithms to pack rectangles into several strips. *Discrete Mathematics and Applications*, 16(1):73–85, 2006.