Improved Pseudo-Polynomial-Time Approximation for Strip Packing*

Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, and Arindam Khan

IDSIA, USI-SUPSI, Switzerland
`[waldo,fabrizio,salvatore,arindam]@idsia.ch`

Abstract

We study the *strip packing* problem, a classical packing problem which generalizes both *bin packing* and *makespan minimization*. Here we are given a set of axis-parallel rectangles in the two-dimensional plane and the goal is to pack them in a vertical strip of fixed width such that the height of the obtained packing is minimized. The packing must be *non-overlapping* and the rectangles cannot be *rotated*.

A reduction from the *partition* problem shows that no approximation better than 3/2 is possible for strip packing in polynomial time (assuming P≠NP). Nadiradze and Wiese [SODA16] overcame this barrier by presenting a $(\frac{7}{5}+\epsilon)$-approximation algorithm in pseudo-polynomial-time (PPT). As the problem is strongly NP-hard, it does not admit an exact PPT algorithm (though a PPT approximation scheme might exist).

In this paper we make further progress on the PPT approximability of strip packing, by presenting a $(\frac{4}{3}+\epsilon)$-approximation algorithm. Our result is based on a non-trivial repacking of some rectangles in the *empty space* left by the construction by Nadiradze and Wiese, and in some sense pushes their approach to its limit.

Our PPT algorithm can be adapted to the case where we are allowed to rotate the rectangles by 90°, achieving the same approximation factor and breaking the polynomial-time approximation barrier of 3/2 for the case with rotations as well.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases approximation algorithms, strip packing, rectangle packing, cutting-stock problem.

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.

1 Introduction

In this paper, we consider the *strip packing* problem, a well-studied classical two-dimensional packing problem [6, 14, 28]. Here we are given a collection of rectangles, and an infinite vertical strip of width W in the two dimensional (2-D) plane. We need to find an axis-parallel embedding of the rectangles *without rotations* inside the strip so that no two rectangles overlap (feasible *packing*). Our goal is to minimize the total height of this packing.

More formally, we are given a parameter $W \in \mathbb{N}$ and a set $\mathcal{R} = \{R_1, \ldots, R_n\}$ of rectangles, each one characterized by a width $w_i \in \mathbb{N}$, $w_i \leq W$, and a height $h_i \in \mathbb{N}$. A packing of $\mathcal{R}$ is a pair $(x_i, y_i) \in \mathbb{N} \times \mathbb{N}$ for each R_i, with $0 \leq x_i \leq W - w_i$, meaning that the left-bottom corner of R_i is placed in position (x_i, y_i) and its right-top corner in position

* Partially supported by ERC Starting Grant NEWNET 279352 and SNSF Grant APXNET 200021_159697/1.

$(x_i + w_i, y_i + h_i)$. This packing is feasible if the interior of rectangles is disjoint in this embedding (or equivalently rectangles are allowed to overlap on their boundary only). Our goal is to find a feasible packing of minimum *height* $\max_i\{y_i + h_i\}$.

Strip packing is a natural generalization of *one-dimensional bin packing* [13] (when all the rectangles have the same height) and *makespan minimization* [12] (when all the rectangles have the same width). The problem has lots of applications in industrial engineering and computer science, specially in cutting stock, logistics and scheduling [28, 20]. Recently, there have been a lot of applications of strip packing in electricity allocation and peak demand reductions in smart-grids [36, 27, 32].

A simple reduction from the *partition* problem shows that the problem cannot be approximated within a factor $\frac{3}{2} - \varepsilon$ for any $\varepsilon > 0$ in polynomial-time unless P=NP. This reduction relies on exponentially large (in n) rectangle widths.

Let $OPT = OPT(\mathcal{R})$ denote the optimal height for the considered strip packing instance $(\mathcal{R}, W)$, and $h_{\max} = h_{\max}(\mathcal{R})$ (resp. $w_{\max} = w_{\max}(\mathcal{R})$) be the largest height (resp. width) of any rectangle in $\mathcal{R}$. Observe that trivially $OPT \geq h_{\max}$. W.l.o.g. we can assume that $W \leq nw_{max}$. The first non-trivial approximation algorithm for strip packing, with approximation ratio 3, was given by Baker, Coffman and Rivest [6]. The First-Fit-Decreasing-Height algorithm (FFDH) by Coffman et al. [14] gives a 2.7 approximation. Sleator [34] gave an algorithm that generates packing of height $2OPT + \frac{h_{max}}{2}$, hence achieving a 2.5 approximation. Afterwards, Steinberg [35] and Schiermeyer [33] independently improved the approximation ratio to 2. Harren and van Stee [21] first broke the barrier of 2 with their 1.9396 approximation. The present best $(\frac{5}{3} + \varepsilon)$-approximation is due to Harren et al. [20].

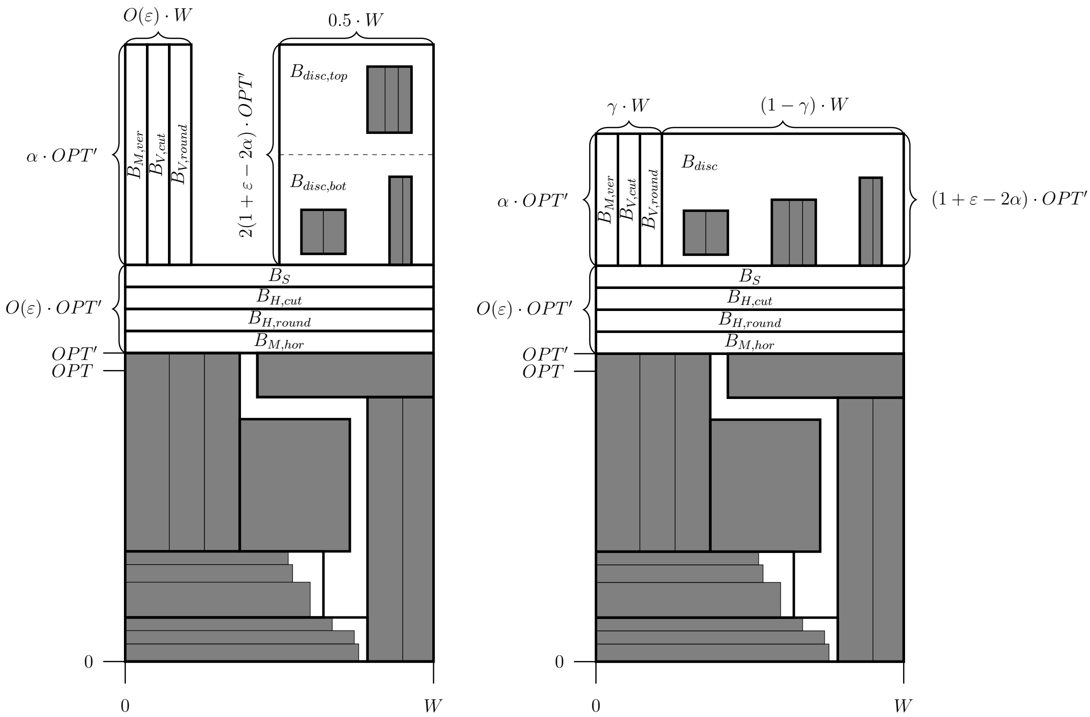
Nadiradze and Wiese [31] overcame the $\frac{3}{2}$-inapproximability barrier by presenting a $(\frac{7}{5} + \epsilon)$-approximation algorithm running in pseudo-polynomial-time (PPT). More specifically, the running time of their algorithm is $O((Nn)^{O(1)})$, where $N = \max\{w_{max}, h_{max}\}$[1]. As strip packing is strongly NP-hard [17], it does not admit an exact PPT algorithm. However, the existence of a PPT approximation scheme is currently not excluded.

1.1 Our contribution and techniques

In this paper, we make progress on the PPT approximability of strip packing, by presenting an improved $(\frac{4}{3} + \varepsilon)$ approximation. Our approach refines the technique of Nadiradze and Wiese [31], that modulo several technical details works as follows. Let $\alpha \in [1/3, 1/2)$ be a proper constant parameter, and define a rectangle R_i to be *tall* if $h_i > \alpha \cdot OPT$. They prove that the optimal packing can be structured into a constant number of axis-aligned rectangular regions (*boxes*), that occupy a total height of $OPT' \leq (1 + \varepsilon)OPT$ inside the vertical strip. Some rectangles are not fully contained into one box (they are *cut* by some box). Among them, tall rectangles remain in their original position. All the other cut rectangles are repacked on top of the boxes: part of them in a horizontal box of size $W \times O(\varepsilon)OPT$, and the remaining ones in a vertical box of size $O(\varepsilon W) \times \alpha\, OPT$ (that we next imagine as placed on the top-left of the packing under construction).

Some of these boxes contain only relatively high rectangles (including tall ones) of relatively small width. The next step is a rearrangement of the rectangles inside one such *vertical* box $\overline{B}$ (see Figure 3a), say of size $\overline{w} \times \overline{h}$: they first slice non-tall rectangles into unit width rectangles (this slicing can be finally avoided with standard techniques). Then

[1] For the case without rotations, the polynomial dependence on h_{max} can indeed be removed with standard techniques.



(a) Final packing obtained by Nadiradze & Wiese [31].

(b) Final packing obtained in this work. Here γ is a small constant depending on ε.

Figure 1 Comparison of final solutions.

they shift tall rectangles to the top/bottom of $\overline{B}$, shifting sliced rectangles consequently (see Figure 3b). Now they discard all the (sliced) rectangles completely contained in a central horizontal region of size $\overline{w}\times(1+\varepsilon-2\alpha)\overline{h}$, and they *nicely rearrange* the remaining rectangles into a constant number of ***sub-boxes*** (excluding possibly a few more non-tall rectangles, that can be placed in the additional vertical box).

These discarded rectangles can be packed into 2 extra boxes of size $\frac{\overline{w}}{2}\times(1+\varepsilon-2\alpha)\overline{h}$ (see Figure 3d). In turn, the latter boxes can be packed into two ***discarded*** boxes of size $\frac{W}{2}\times(1+\varepsilon-2\alpha)OPT'$, that we can imagine as placed, one on top of the other, on the top-right of the packing. See Figure 1a for an illustration of the final packing. This leads to a total height of $(1+\max\{\alpha, 2(1-2\alpha)\}+O(\varepsilon))\cdot OPT$, which is minimized by choosing $\alpha=\frac{2}{5}$.

Our main technical contribution is a repacking lemma that allows one to repack a small fraction of the discarded rectangles of a given box inside the free space left by the corresponding sub-boxes (while still having $O_\varepsilon(1)$ many sub-boxes in total). This is illustrated in Figure 3e. This way we can pack all the discarded rectangles into a *single* discarded box of size $(1-\gamma)W\times(1+\varepsilon-2\alpha)OPT'$, where γ is a small constant depending on ε, that we can place on the top-right of the packing. The vertical box where the remaining rectangles are packed still fits to the top-left of the packing, next to the discarded box. See Figure 1b for an illustration. Choosing $\alpha=1/3$ gives the claimed approximation factor.

We remark that the basic approach by Nadiradze and Wiese strictly requires that at most 2 tall rectangles can be packed one on top of the other in the optimal packing, hence imposing $\alpha\geq 1/3$. Thus in some sense we pushed their approach to its limit.

The algorithm by Nadiradze and Wiese [31] is not directly applicable to the case when $90°$ rotations are allowed. In particular, they use a linear program to pack some rectangles. When rotations are allowed, it is unclear how to decide which rectangles are packed by the linear program. We use a combinatorial *container*-based approach to circumvent this limitation, which allows us to pack all the rectangles using dynamic programming. This way we achieve a PPT $(4/3+\varepsilon)$-approximation for strip packing with rotations, breaking the polynomial-time approximation barrier of $3/2$ for that variant as well.

1.2 Related work

For packing problems, many pathological lower bound instances occur when OPT is small. Thus it is often insightful to consider the *asymptotic approximation ratio*. Coffman et al. [14] described two *level-oriented* algorithms, Next-Fit-Decreasing-Height (NFDH) and First-Fit-Decreasing-Height (FFDH), that achieve asymptotic approximations of 2 and 1.7, respectively. After a sequence of improvements [18, 5], the seminal work of Kenyon and Rémila [28] provided an asymptotic polynomial-time approximation scheme (APTAS) with an additive term $O\left(\frac{h_{max}}{\varepsilon^2}\right)$. The latter additive term was subsequently improved to h_{max} by Jansen and Solis-Oba [24].

In the variant of strip packing *with rotations*, we are allowed to rotate the input rectangles by $90°$ (in other terms, we are free to swap the width and height of an input rectangle). The case with rotations is much less studied in the literature. It seems that most techniques that work for the case without rotations can be extended to the case with rotations, however this is not always a trivial task. In particular, it is not hard to achieve a $2+\varepsilon$ approximation, and the $3/2$ hardness of approximation extends to this case as well [24]. In terms of asymptotic approximation, Miyazawa and Wakabayashi [30] gave an algorithm with asymptotic performance ratio of 1.613. Later, Epstein and van Stee [16] gave a $\frac{3}{2}$ asymptotic approximation. Finally, Jansen and van Stee [25] achieved an APTAS for the case with rotations.

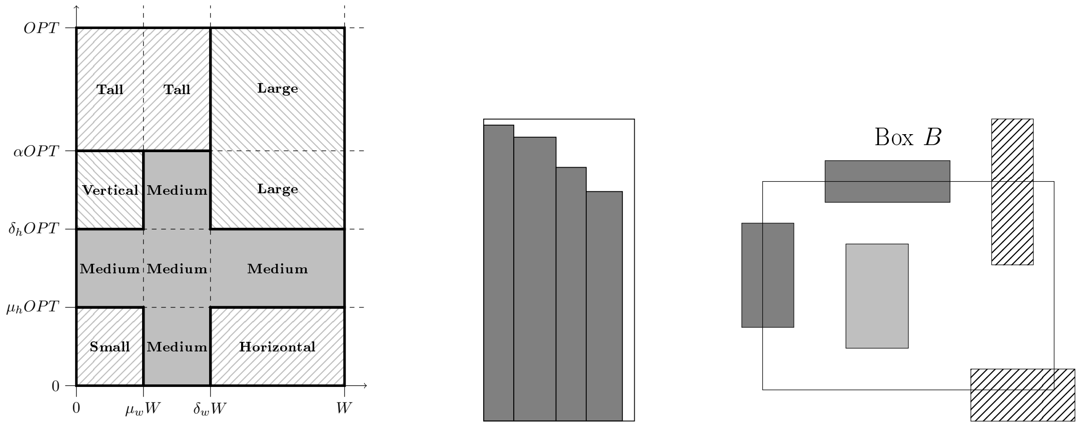
Strip packing has also been well studied for higher dimensions. The present best asymptotic approximation for 3-D strip packing is due to Jansen and Prädel [23] who gave 1.5-approximation extending techniques from 2-D bin packing.

There are many other related geometric packing problems. For example, in the *independent set of rectangles* problems we are given a collection of axis-parallel rectangles embedded in the plane, and we need to find a maximum cardinality/weight subset of non-overlapping rectangles [1, 10, 11]. Interesting connections between this problems and *unsplittable flow on a path* were recently discovered [3, 4, 7, 9, 19]. In the *geometric knapsack* problem we wish to pack a maximum cardinality/profit subset of the rectangles in a given square knapsack [2, 26]. One can also consider a natural geometric version of bin packing, where one needs to pack a given set of rectangles in the smallest possible number of square bins [8]. We refer the readers to [29] for a survey on geometric packing problems.

1.3 Organization of the paper

First, we discuss some preliminaries and notations in Section 2. Section 3 contains our main technical contribution, our *repacking lemma*. There we also discuss a refined structural result leading to a packing into $O_\varepsilon(1)$ many *containers*. In Section 4, we describe our algorithm to pack the rectangles. Then in Section 5, we extend our algorithm to the case with rotations. Finally, in Section 6, we conclude with some observations.

Due to space constraints, some proofs are omitted from this extended abstract and will appear in the full version of the paper.



(a) Each rectangle is represented as a point on the plane with x (resp., y) coordinate indicating its width (resp., height).

(b) Example of *vertical container*. Every vertical line intersects at most one rectangle.

(c) Gray rectangles are *nicely cut* by B; dashed rectangles are *cut* but *not nicely cut* by B, and light gray rectangle is not cut by B.

Figure 2 Illustration of some of the definitions used in this paper.

2 Preliminaries and notations

Throughout the present work, we will follow the notation from [31], which will be explained as it is needed.

Recall that $OPT \in \mathbb{N}$ denotes the height of the optimal packing for instance $\mathcal{R}$. By trying all the pseudo-polynomially many possibilities, we can assume that OPT is known to the algorithm. Given a set $\mathcal{M} \subseteq \mathcal{R}$ of rectangles, $a(\mathcal{M})$ will denote the total area of rectangles in $\mathcal{M}$, i.e., $a(\mathcal{M}) = \sum_{R_i \in \mathcal{M}} h_i \cdot w_i$, and $h_{\max}(\mathcal{M})$ (resp. $w_{\max}(\mathcal{M})$) denotes the maximum height (resp. width) of rectangles in $\mathcal{M}$. Throughout this work, a *box* of size $a \times b$ means an axis-aligned rectangular region of width a and height b.

In order to lighten the notation, we sometimes interpret a rectangle/box as the corresponding region inside the strip according to some given embedding. The latter embedding will not be specified when clear from the context. Similarly, we sometimes describe an embedding of some rectangles inside a box, and then embed the box inside the strip: the embedding of the considered rectangles is shifted consequently in that case.

A vertical (resp. horizontal) *container* is an axis-aligned rectangular region where we implicitly assume that rectangles are packed one next to the other from left to right (resp., bottom to top), i.e., any vertical (resp. horizontal) line intersects only one packed rectangle (see Figure 2b). Container-like packings will turn out to be particularly useful since they naturally induce a (one-dimensional) knapsack instance.

2.1 Classification of rectangles

Let $0 < \varepsilon < \alpha$, and assume for simplicity that $\frac{1}{\varepsilon} \in \mathbb{N}$. We first classify the input rectangles into six groups according to parameters $\delta_h, \delta_w, \mu_h, \mu_w$ satisfying $\varepsilon \geq \delta_h > \mu_h > 0$ and $\varepsilon \geq \delta_w > \mu_w > 0$, whose values will be chosen later (see also Figure 2a). A rectangle R_i is

- *Large* if $h_i \geq \delta_h OPT$ and $w_i \geq \delta_w W$.
- *Tall* if $h_i > \alpha OPT$ and $w_i < \delta_w W$.
- *Vertical* if $h_i \in [\delta_h OPT, \alpha OPT]$ and $w_i \leq \mu_w W$,

- *Horizontal* if $h_i \leq \mu_h OPT$ and $w_i \geq \delta_w W$,
- *Small* if $h_i \leq \mu_h OPT$ and $w_i \leq \mu_w W$;
- *Medium* in all the remaining cases, i.e., if $h_i \in (\mu_h OPT, \delta_h OPT)$, or $w_i \in (\mu_w W, \delta_w W)$ and $h_i \leq \alpha OPT$.

We use L, T, V, H, S, and M to denote large, tall, vertical, horizontal, small, and medium rectangles, respectively. We remark that, differently from [31], we need to allow $\delta_h \neq \delta_w$ and $\mu_h \neq \mu_w$ due to some additional constraints in our construction (See Section 4).

Notice that according to this classification, every vertical line across the optimal packing intersects at most two tall rectangles. The following lemma allows us to choose $\delta_h, \delta_w, \mu_h$ and μ_w in such a way that δ_h and μ_h (δ_w and μ_w, respectively) differ by a large factor, and medium rectangles have small total area.

▶ **Lemma 1.** *Given a polynomial-time computable function $f : (0,1) \to (0,1)$, with $f(x) < x$, any constant $\varepsilon \in (0,1)$, and any positive integer k, we can compute in polynomial time a set Δ of $T = 2(\frac{1}{\varepsilon})^k$ many positive real numbers upper bounded by ε, such that there is at least one number $\delta_h \in \Delta$ so that $a(M) \leq \varepsilon^k \cdot OPT \cdot W$ by choosing $\mu_h = f(\delta_h)$, $\mu_w = \frac{\varepsilon\mu_h}{12}$, and $\delta_w = \frac{\varepsilon\delta_h}{12}$.*

Function f and constant k will be chosen later. From now on, assume that $\delta_h, \delta_w, \mu_h$ and μ_w are chosen according to Lemma 1.

2.2 Overview of the algorithm

We next overview some of the basic results in [31] that are needed in our result. We define the constant $\gamma := \frac{\varepsilon\delta_h}{2}$, and w.l.o.g. assume $\gamma \cdot OPT \in \mathbb{N}$.

Let us forget for a moment small rectangles S. We will pack all the remaining rectangles $L \cup H \cup T \cup V \cup M$ into a sufficiently small number of boxes embedded into the strip. By standard techniques, as in [31], it is then possible to pack S (essentially using NFDH in a proper grid defined by the above boxes) while increasing the total height at most by $O(\varepsilon)OPT$. See Section 4.1 for more details on packing of small rectangles.

The following lemma from [31] allows one to round the heights and positions of rectangles of large enough height, without increasing much the height of the packing.

▶ **Lemma 2.** *[31] There exists a feasible packing of height $OPT' \leq (1+\varepsilon)OPT$ where: (1) the height of each rectangles in $L \cup T \cup V$ is rounded up to the closest integer multiple of $\gamma \cdot OPT$ and (2) their x-coordinates are as in the optimal solution and their y-coordinates are integer multiples of $\gamma \cdot OPT$.*

We next focus on rounded rectangle heights (i.e., implicitly replace $L \cup T \cup V$ by their rounded version) and on this slightly suboptimal solution of height OPT'.

The following lemma helps us to pack rectangles in M.

▶ **Lemma 3.** *If k in Lemma 1 is chosen sufficiently large, all the rectangles in M can be packed in polynomial time into a box $B_{M,hor}$ of size $W \times O(\varepsilon)OPT$ and a box $B_{M,ver}$ of size $(\frac{\gamma}{3}W) \times (\alpha OPT)$. Furthermore, there is one such packing using $\frac{3\varepsilon}{\mu_h}$ vertical containers in $B_{M,hor}$ and $\frac{\gamma}{3\mu_w}$ horizontal containers in $B_{M,ver}$.*

We say that a rectangle R_i is *cut* by a box B if both $R_i \setminus B$ and $B \setminus R_i$ are non-empty (considering both R_i and B as open regions with an implicit embedding on the plane). We say that a rectangle $R_i \in H$ (resp. $R_i \in T \cup V$) is *nicely cut* by a box B if R_i is cut by B and their intersection is a rectangular region of width w_i (resp. height h_i). Intuitively, this means that an edge of B cuts R_i along its longest side (see Figure 2c).

Now it remains to pack $L \cup H \cup T \cup V$: The following lemma, taken from [31] modulo minor technical adaptations, describes an almost optimal packing of those rectangles.

▶ **Lemma 4.** *There is an integer $K_B = (\frac{1}{\varepsilon})(\frac{1}{\delta_w})^{O(1)}$ such that, assuming $\mu_h \leq \frac{\varepsilon\delta_w}{K_B}$, there is a partition of the region $B_{OPT'} := [0, W] \times [0, OPT']$ into a set $\mathcal{B}$ of at most K_B boxes and a packing of the rectangles in $L \cup T \cup V \cup H$ such that:*

- *each box has size equal to the size of some $R_i \in L$ (*large box*), or has height at most $\delta_h OPT'$ (*horizontal box*), or has width at most $\delta_w W$ (*vertical box*);*
- *each $R_i \in L$ is contained into a large box of the same size;*
- *each $R_i \in H$ is contained into a horizontal box or is cut by some box. Furthermore, the total area of horizontal cut rectangles is at most $W \cdot O(\varepsilon)OPT'$;*
- *each $R_i \in T \cup V$ is contained into a vertical box or is nicely cut by some vertical box.*

We denote the sets of vertical, horizontal, and large boxes by $\mathcal{B}_V, \mathcal{B}_H$ and $\mathcal{B}_L$, respectively. Observe that $\mathcal{B}$ can be guessed in PPT. We next use $T_{cut} \subseteq T$ and $V_{cut} \subseteq V$ to denote tall and vertical cut rectangles in the above lemma, respectively. Let us also define $T_{box} = T \setminus T_{cut}$ and $V_{box} = V \setminus V_{cut}$.

Using standard techniques (see e.g. [31]), we can pack all the rectangles excluding the ones contained in vertical boxes in a convenient manner. This is summarized in the following lemma.

▶ **Lemma 5.** *Given $\mathcal{B}$ as in Lemma 4 and assuming $\mu_w \leq \frac{\gamma\delta_h}{6K_B(1+\varepsilon)}$, there exists a packing of $L \cup H \cup T \cup V$ such that:*

1. *all the rectangles in L are packed in $\mathcal{B}_L$;*
2. *all the rectangles in H are packed in $\mathcal{B}_H$ plus an additional box $B_{H,cut}$ of size $W \times O(\varepsilon)OPT$;*
3. *all the rectangles in $T_{cut} \cup T_{box} \cup V_{box}$ are packed as in Lemma 4;*
4. *all the rectangles in V_{cut} are packed in an additional vertical box $B_{V,cut}$ of size $(\frac{\gamma}{3}W) \times (\alpha OPT)$.*

We will pack all the rectangles (essentially) as in [31], with the exception of $T_{box} \cup V_{box}$ where we exploit a refined approach. This is the technical heart of this paper, and it is discussed in the next section.

3 A repacking lemma

We next describe how to pack rectangles in $T_{box} \cup V_{box}$. In order to highlight our contribution, we first describe how the approach by Nadiradze and Wiese [31] works.

It is convenient to assume that all the rectangles in V_{box} are sliced vertically into sub-rectangles of width 1 each[2]. Let V_{sliced} be such *sliced* rectangles. We will show how to pack all the rectangles in $T_{box} \cup V_{sliced}$ into a constant number of sub-boxes. Using standard techniques it is then possible to pack V_{box} into the space occupied by V_{sliced} plus an additional box $B_{V,round}$ of size $(\frac{\gamma}{3}W) \times \alpha OPT$.

We next focus on a specific vertical box $\overline{B}$, say of size $\overline{w} \times \overline{h}$ (see Figure 3a). Let $\overline{T}_{cut}$ be the tall rectangles cut by $\overline{B}$. Observe that there are at most 4 such rectangles (2 on the left/right side of $\overline{B}$). The rectangles in $\overline{T}_{cut}$ are packed as in Lemma 5. Let also $\overline{T}$ and $\overline{V}$ be the tall rectangles and sliced vertical rectangles, respectively, originally packed completely inside $\overline{B}$.

[2] For technical reasons, slices have width 1/2 in [31]. For our algorithm, slices of width 1 suffice.

They show that it is possible to pack $\overline{T} \cup \overline{V}$ into a constant size set $\overline{\mathcal{S}}$ of sub-boxes contained inside $\overline{B} - \overline{T}_{cut}$, plus an additional box $\overline{D}$ of size $\overline{w} \times (1+\varepsilon-2\alpha)\overline{h}$. Here $\overline{B} - \overline{T}_{cut}$ denotes the region inside $\overline{B}$ not contained in $\overline{T}_{cut}$. In more detail, they start by considering each rectangle $R_i \in \overline{T}$. Since $\alpha \geq \frac{1}{3}$ by assumption, one of the regions above or below R_i cannot contain another tall rectangle in $\overline{T}$, say the first case applies (the other one being symmetric). Then we move R_i up so that its top side overlaps with the top side of $\overline{B}$. The sliced rectangles in $\overline{V}$ that are covered this way are shifted right below R (note that there is enough free space by construction). At the end of the process all the rectangles in $\overline{T}$ touch at least one of the top and bottom side of $\overline{B}$ (see Figure 3b). Note that no rectangle is discarded up to this point.

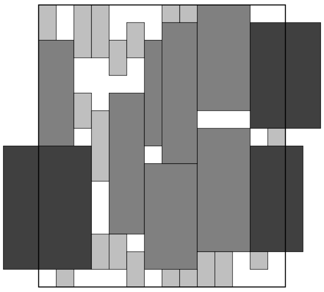
Next, we partition the space inside $\overline{B}-(\overline{T}\cup\overline{T}_{cut})$ into maximal height unit-width vertical stripes. We call each such stripe a *free rectangle* if both its top and bottom side overlap with the top or bottom side of some rectangle in $\overline{T} \cup \overline{T}_{cut}$, and otherwise a *pseudo rectangle* (see Figure 3c). We define the i-th free rectangle to be the free rectangle contained in stripe $[i-1, i] \times [0, \overline{h}]$.

Note that all the free rectangles are contained in a horizontal region of width $\overline{w}$ and height at most $\overline{h} - 2\alpha OPT \leq \overline{h} - 2\alpha \frac{OPT'}{1+\varepsilon} \leq \overline{h}(1 - \frac{2\alpha}{1+\varepsilon}) \leq \overline{h}(1+\varepsilon-2\alpha)$ contained in the central part of $\overline{B}$. Let $\overline{V}_{disc}$ be the set of (sliced vertical) rectangles contained in the free rectangles. Rectangles in $\overline{V}_{disc}$ can be obviously packed inside $\overline{D}$. For each corner Q of the box $\overline{B}$, we consider the maximal rectangular region that has Q as a corner and only contains pseudo rectangles whose top/bottom side overlaps with the bottom/top side of a rectangle in $\overline{T}_{cut}$; there are at most 4 such non-empty regions, and for each of them we define a *corner sub-box*, and we call the set of such sub-boxes $\overline{B}_{corn}$ (see Figure 3c). The final step of the algorithm is to rearrange horizontally the pseudo/tall rectangles so that pseudo/tall rectangles of the same height are grouped together *as much as possible* (modulo some technical details). The rectangles in $\overline{B}_{corn}$ are not moved. The *sub-boxes* are induced by maximal consecutive subsets of pseudo/tall rectangles of the same height touching the top (resp., bottom) side of $\overline{B}$ (see Figure 3d). We crucially remark that, by construction, the height of each sub-box (and of $\overline{B}$) is a multiple of γOPT.

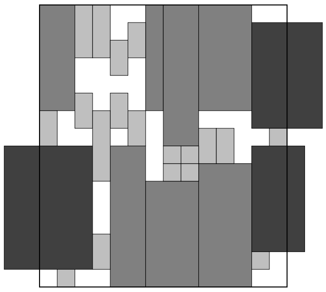
By splitting each discarded box $\overline{D}$ into two halves $\overline{B}_{disc,top}$ and $\overline{B}_{disc,bot}$, and replicating the packing of boxes inside $B_{OPT'}$, it is possible to pack all the discarded boxes into two boxes $B_{disc,top}$ and $B_{disc,bot}$, both of size $\frac{W}{2} \times (1+\varepsilon-2\alpha)OPT'$.

A feasible packing of boxes (and hence of the associated rectangles) of height $(1 + \max\{\alpha, 2(1-2\alpha)\} + O(\varepsilon))OPT$ is then obtained as follows. We first pack $B_{OPT'}$ at the base of the strip, and then on top of it we pack $B_{M,hor}$, two additional boxes $B_{H,round}$ and $B_{H,cut}$ (which will be used to repack the horizontal items), and a box B_S (which will be used to pack some of the small items). The latter 4 boxes all have width W and height $O(\varepsilon OPT')$. On the top right of this packing we place $B_{disc,top}$ and $B_{disc,bot}$, one on top of the other. Finally, we pack $B_{M,ver}$, $B_{V,cut}$ and $B_{V,round}$ on the top left, one next to the other. See Figure 1a for an illustration. The height is minimized for $\alpha = \frac{2}{5}$, leading to a $7/5 + O(\varepsilon)$ approximation.

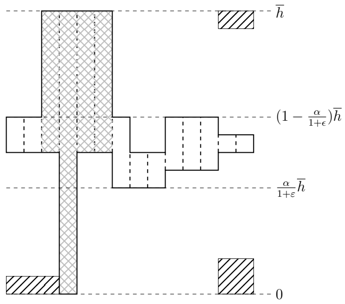
The main technical contribution of this paper is to show how it is possible to repack a subset of $\overline{V}_{disc}$ into the *free* space inside $\overline{B}_{cut} := \overline{B} - \overline{T}_{cut}$ not occupied by sub-boxes, so that the residual sliced rectangles can be packed into a single discarded box $\overline{B}_{disc}$ of size $(1-\gamma)\overline{w} \times (1+\varepsilon-2\alpha)\overline{h}$ (*repacking lemma*). See Figure 3e. This apparently minor saving is indeed crucial: with the same approach as above all the discarded sub-boxes $\overline{B}_{disc}$ can be packed into a single *discarded box* B_{disc} of size $(1-\gamma)W \times (1+\varepsilon-2\alpha)OPT'$. Therefore, we can pack all the previous boxes as before, and B_{disc} on the top right. Indeed,



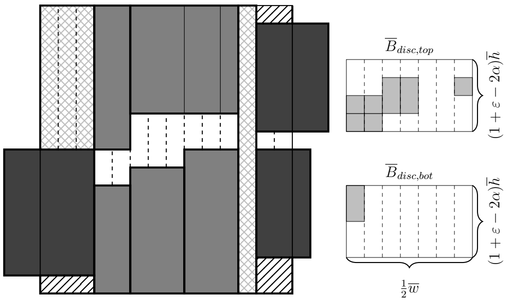
(a) Original packing in a vertical box $\overline{B}$ after removing V_{cut}. Gray rectangles correspond to $\overline{T}$, dark gray rectangles to $\overline{T}_{cut}$ and light gray rectangles to $\overline{V}$.



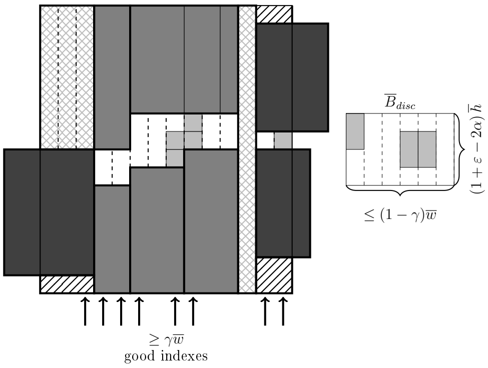
(b) Rectangles in $\overline{T}$ are shifted vertically so that they touch either the top or the bottom of box $\overline{B}$, shifting also slices in $\overline{V}$ accordingly.



(c) We define pseudo rectangles and free space in $\overline{B} - (\overline{T} \cup \overline{T}_{cut})$. Crosshatched stripes correspond to pseudo rectangles, empty stripes to free rectangles, and dashed regions correspond to corner sub-boxes.



(d) Rearrangement of pseudo and tall rectangles to get $O_\varepsilon(1)$ sub-boxes, and additional packing of $\overline{V}_{disc}$ as in [31].



(e) Our refined repacking of $\overline{V}_{disc}$ according to Lemma 6: some vertical slices are repacked in the free space.

Figure 3 Creation of pseudo rectangles, how to get constant number of sub-boxes and repacking of vertical slices in a vertical box $\overline{B}$.

the total width of $B_{M,ver}$, $B_{V,cut}$ and $B_{V,round}$ is at most γW for a proper choice of the parameters. See Figure 1b for an illustration. Altogether the resulting packing has height $(1 + \max\{\alpha, 1 - 2\alpha\} + O(\varepsilon))OPT$. This is minimized for $\alpha = \frac{1}{3}$, leading to the claimed $4/3 + O(\varepsilon)$ approximation.

It remains to prove our repacking lemma.

▶ **Lemma 6** (Repacking Lemma)**.** *Consider a partition of $\overline{D}$ into $\overline{w}$ unit-width vertical stripes. There is a subset of at least $\gamma\overline{w}$ such stripes so that the corresponding sliced vertical rectangles $\overline{V}_{repack}$ can be repacked inside $\overline{B}_{cut} = \overline{B} - \overline{T}_{cut}$ in the space not occupied by sub-boxes.*

Proof. Let $f(i)$ denote the height of the i-th free rectangle, where for notational convenience we introduce a degenerate free rectangle of height $f(i) = 0$ whenever the stripe $[i-1, i] \times [0, \overline{h}]$ inside $\overline{B}$ does not contain any free rectangle. This way we have precisely $\overline{w}$ free rectangles. We remark that free rectangles are defined before the horizontal rearrangement of tall/pseudo rectangles, and the consequent definition of sub-boxes.

Recall that sub-boxes contain tall and pseudo rectangles. Now consider the area in $\overline{B}_{cut}$ not occupied by sub-boxes. Note that this area is contained in the central region of height $\overline{h}(1-\frac{2\alpha}{1+\varepsilon})$. Partition this area into maximal-height unit-width vertical stripes as before (*newly free rectangles*). Let $g(i)$ be the height of the i-th newly free rectangle, where again we let $g(i)=0$ if the stripe $[i-1,i]\times[0,\overline{h}]$ does not contain any (positive area) free region. Note that, since tall and pseudo rectangles are only shifted horizontally in the rearrangement, it must be the case that:

$$\sum_{i=1}^{\overline{w}} f(i) = \sum_{i=1}^{\overline{w}} g(i).$$

Let G be the (*good*) indexes where $g(i)\geq f(i)$, and $\overline{G}=\{1,\dots,\overline{w}\}-G$ be the *bad* indexes with $g(i)<f(i)$. Observe that for each $i\in G$, it is possible to pack the i-th free rectangle inside the i-th newly free rectangle, therefore freeing a unit-width vertical strip inside $\overline{D}$. Thus it is sufficient to show that $|G|\geq\gamma\overline{w}$.

Observe that, for $i\in\overline{G}$, $f(i)-g(i)\geq\gamma OPT\geq\gamma\frac{\overline{h}}{1+\varepsilon}$: indeed, both $f(i)$ and $g(i)$ must be multiples of γOPT since they correspond to the height of $\overline{B}$ minus the height of one or two tall/pseudo rectangles. On the other hand, for any index i, $g(i)-f(i)\leq g(i)\leq(1-\frac{2\alpha}{1+\varepsilon})\overline{h}$, by the definition of g. Altogether

$$(1-\frac{2\alpha}{1+\varepsilon})\overline{h}\cdot|G|\geq\sum_{i\in G}(g(i)-f(i))=\sum_{i\in\overline{G}}(f(i)-g(i))\geq\frac{\gamma\overline{h}}{1+\varepsilon}\cdot|\overline{G}|=\frac{\gamma\overline{h}}{1+\varepsilon}\cdot(\overline{w}-|G|)$$

We conclude that $|G|\geq\frac{\gamma}{1+\varepsilon-2\alpha+\gamma}\overline{w}$. The claim follows since by assumption $\alpha>\varepsilon\geq\gamma$. ◀

The original algorithm in [31] use standard LP-based techniques, as in [28], to pack the horizontal rectangles. We can avoid that via a refined structural lemma: here boxes and sub-boxes are further partitioned into vertical (resp., horizontal) containers. Rectangles are then packed into such containers as mentioned earlier: one next to the other from left to right (resp., bottom to top). Containers define a multiple knapsack instance, that can be solved optimally in PPT via dynamic programming. This approach has two main advantages:

- It leads to a simpler algorithm.
- It can be easily adapted to the case with rotations, as discussed in Section 5.

We omit the proof of the following Lemma.

▶ **Lemma 7.** *By choosing $\alpha=1/3$, there is an integer $K_F\leq\left(\frac{1}{\varepsilon\delta_w}\right)^{O(1/(\delta_w\varepsilon))}$ such that, assuming $\mu_h\leq\frac{\varepsilon}{K_F}$ and $\mu_w\leq\frac{\gamma}{3K_F}$, there is a packing of $\mathcal{R}\setminus S$ in the region $[0,W]\times[0,(4/3+O(\varepsilon))OPT']$ with the following properties:*

- *All the rectangles in $\mathcal{R}\setminus S$ are contained in $K_{TOTAL}=O_\varepsilon(1)$ horizontal or vertical containers, such that each of these containers is either contained in or disjoint from $\mathcal{B}_{OPT'}$;*
- *At most K_F containers are contained in $\mathcal{B}_{OPT'}$, and their total area is at most $a(\mathcal{R}\setminus S)$.*

4 A refined algorithm

First of all, we find $\mu_h,\delta_h,\mu_w,\delta_w$ as required by Lemma 1; this way, we can find the set S of small rectangles. Consider the packing of Lemma 7: all the non-small rectangles are packed into $K_{TOTAL}=O_\varepsilon(1)$ containers, and only K_F of them are contained in $B_{OPT'}$. Since their position (x,y) and their size (w,h) are w.l.o.g. contained in $\{0,\dots,W\}\times\{0,\dots,nh_{max}\}$, we can enumerate in PPT over all the possible feasible such packings of $k\leq K_{TOTAL}$ containers, and one of those will coincide with the packing defined by Lemma 7.

Containers naturally induce a multiple knapsack problem: for each horizontal container C_j of size $w_{C_j} \times h_{C_j}$, we create a (one-dimensional) knapsack j of size h_{C_j}. Furthermore, we define the size $b(i,j)$ of rectangle R_i w.r.t. knapsack j as h_i if $h_i \le h_{C_j}$ and $w_i \le w_{C_j}$. Otherwise $b(i,j) = +\infty$ (meaning that R_i does not fit in C_j). The construction for vertical containers is symmetric. This multiple knapsack problem can be easily solved optimally (hence packing all the rectangles) in PPT via dynamic programming.

Note that unlike [31], we do not use linear programming to pack horizontal rectangles, which will be crucial when we extend our approach to the case with rotations.

4.1 Packing the small rectangles

It remains to pack the small rectangles S. We will pack them in the free space left by containers inside $[0,W]\times[0,OPT']$ plus an additional box B_S of small height as the following lemma states. By placing box B_S on top of the remaining packed rectangles, the final height of the solution increases only by $\varepsilon \cdot OPT'$.

▶ **Lemma 8.** *Assuming $\mu_h \le \frac{1}{31K_F^2}$, it is possible to pack in polynomial time all the rectangles in S into the area $[0,W] \times [0,OPT']$ not occupied by containers plus an additional box B_S of size $W \times \varepsilon OPT'$.*

Proof. We first extend the sides of the containers inside $[0,W] \times [0,OPT']$ in order to define a grid. This procedure partitions the free space in $[0,W] \times [0,OPT']$ into a constant number of rectangular regions (at most $(2K_F+1)^2 \le 5K_F^2$ many) whose total area is at least $a(S)$ thanks to Lemma 7. Let $\mathcal{B}_{small}$ be the set of such rectangular regions with width at least $\mu_w W$ and height at least $\mu_h OPT$ (notice that the total area of rectangular regions not in $\mathcal{B}_{small}$ is at most $5K_F^2\mu_w\mu_h \cdot W \cdot OPT$). We now use NFDH to pack a subset of S into the regions in $\mathcal{B}_{small}$. By standard properties of NFDH, since each region in $\mathcal{B}_{small}$ has size at most $W \times OPT'$ and each item in S has width at most $\mu_w W$ and height at most $\mu_h OPT$, the total area of the unpacked rectangles from S can be bounded above by $5K_F^2 \cdot \left(\mu_w\mu_h WOPT + \mu_h OPT \cdot W + \mu_w W \cdot OPT'\right) \le 15K_F^2\mu_h \cdot OPT' \cdot W$. Therefore we can pack the latter small rectangles with NFDH in an additional box B_S of width W and height $\mu_h OPT + 30K_F^2\mu_h OPT' \le \varepsilon \cdot OPT'$ provided that $\mu_h \le \frac{1}{31K_F^2}$. ◀

The (rather technical) details on how to choose f and k (and consequently the actual values of μ_h, δ_w, and μ_w) will be discussed in the full version of this paper. We next summarize the constraints that arise from the analysis:

- $\mu_w = \frac{\varepsilon\mu_h}{12}$ and $\delta_w = \frac{\varepsilon\delta_h}{12}$ (Lemma 1),
- $\gamma = \frac{\varepsilon\delta_h}{2}$ (Lemma 2),
- $6\varepsilon^k \le \frac{\gamma}{6}$ (Lemma 3)
- $\mu_h \le \frac{\varepsilon\delta_w}{K_B}$ (Lemma 4),
- $\mu_w \le \gamma\frac{\delta_h}{6K_B(1+\varepsilon)}$ (Lemma 5),
- $\mu_w \le \frac{\gamma}{3K_F}$ (Lemma 7),
- $\mu_h \le \frac{\varepsilon}{K_F}$ (Lemma 7),
- $\mu_h \le \frac{1}{31K_F^2}$ (Lemma 8)

It is not difficult to see that all the constraints are satisfied by choosing $f(x) = (\varepsilon x)^{C/(\varepsilon x)}$ for a large enough constant C and $k = \left\lceil \log_\varepsilon\left(\frac{\gamma}{36}\right)\right\rceil$. Finally we achieve the claimed result.

▶ **Theorem 9.** *There is a PPT $(\frac{4}{3}+\varepsilon)$-approximation algorithm for strip packing.*

5 Extension to the case with rotations

In this section, we briefly explain the changes needed in the above algorithm for the case with rotations.

We first observe that, by considering the rotation of rectangles as in the optimum solution, Lemma 7 still applies (for a proper choice of the parameters, that can be guessed). Therefore we can define a multiple knapsack instance, where knapsack sizes are defined as before. Some extra care is needed to define the size $b(i, j)$ of rectangle R_i into a container C_j of size $w_{C_j} \times h_{C_j}$. Assume C_j is horizontal, the other case being symmetric. If rectangle R_i fits in C_j both rotated and non-rotated, then we set $b(i, j) = \min\{w_i, h_i\}$ (this dominates the size occupied in the knapsack by the optimal rotation of R_i). If R_i fits in C_j only non-rotated (resp., rotated), we set $b(i, j) = h_i$ (resp., $b(i, j) = w_i$). Otherwise we set $b(i, j) = +\infty$.

There is a final difficulty that we need to address: we can not say a priori whether a rectangle is small (and therefore should be packed in the final stage). To circumvent this difficulty, we define one extra knapsack k' whose size is the total area in $\mathcal{B}_{OPT'}$ not occupied by the containers. The size $b(i, k')$ of R_i in this knapsack is the area $a(R_i) = w_i \cdot h_i$ of R_i provided that R_i or its rotation by $90°$ is small w.r.t. the current choice of the parameters $(\delta_h, \mu_h, \delta_w, \mu_w)$. Otherwise $b(i, k') = +\infty$.

By construction, the above multiple knapsack instance admits a feasible solution that packs all the rectangles. This immediately implies a packing of all the rectangles, excluding the (small) ones in the extra knapsack. Those rectangles can be packed using NFDH as in the proof of Lemma 8 (here however we must choose a rotation such that the considered rectangle is small). Altogether we achieve:

▶ **Theorem 10.** *There is a PPT $(\frac{4}{3} + \varepsilon)$-approximation algorithm for strip packing with rotations.*

6 Conclusions

In this paper we obtained a PPT $4/3+\varepsilon$ approximation for strip packing (with and without rotations). Our approach refines and, in some sense, pushes to its limit the basic approach in previous work by Nadiradze and Wiese [31]. Indeed, the rearrangement of rectangles inside a box crucially exploits the fact that there are at most 2 tall rectangles packed on top of each other in the optimal packing, hence requiring $\alpha \geq 1/3$. We believe that any further improvement requires substantially new algorithmic ideas.

A PPT approximation scheme for strip packing is not excluded by the current inapproximability results (essentially, only strong NP-hardness). Note that, like bin packing, strip packing admits an asymptotic polynomial-time approximation scheme (APTAS), and bin packing admits a PPT approximation scheme [22, 15]. It is an interesting open problem to find a PPT approximation scheme for this problem, or to prove some stronger hardness of approximation result in PPT.

References

1 Anna Adamaszek and Andreas Wiese. Approximation schemes for maximum weight independent set of rectangles. In *FOCS*, pages 400–409, 2013.

2 Anna Adamaszek and Andreas Wiese. A quasi-PTAS for the two-dimensional geometric knapsack problem. In *SODA*, pages 1491–1505, 2015.

3 Aris Anagnostopoulos, Fabrizio Grandoni, Stefano Leonardi, and Andreas Wiese. Constant integrality gap LP formulations of unsplittable flow on a path. In *IPCO*, pages 25–36, 2013.

4 Aris Anagnostopoulos, Fabrizio Grandoni, Stefano Leonardi, and Andreas Wiese. A mazing $2+\varepsilon$ approximation for unsplittable flow on a path. In *SODA*, pages 26–41, 2014.

5 Brenda S. Baker, Donna J. Brown, and Howard P. Katseff. A 5/4 algorithm for two-dimensional packing. *Journal of algorithms*, 2(4):348–368, 1981.

6 Brenda S. Baker, Edward G. Coffman Jr., and Ronald L. Rivest. Orthogonal packings in two dimensions. *SIAM J. Comput.*, 9(4):846–855, 1980.

7 Nikhil Bansal, Amit Chakrabarti, Amir Epstein, and Baruch Schieber. A quasi-PTAS for unsplittable flow on line graphs. In *STOC*, pages 721–729, 2006.

8 Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In *SODA*, pages 13–25, 2014.

9 Paul S. Bonsma, Jens Schulz, and Andreas Wiese. A constant-factor approximation algorithm for unsplittable flow on paths. *SIAM J. Comput.*, 43(2):767–799, 2014.

10 Parinya Chalermsook and Julia Chuzhoy. Maximum independent set of rectangles. In *SODA*, pages 892–901, 2009.

11 Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48(2):373–392, 2012.

12 Edward G. Coffman Jr. and John L. Bruno. *Computer and job-shop scheduling theory*. John Wiley & Sons, 1976.

13 Edward G. Coffman Jr., János Csirik, Gábor Galambos, Silvano Martello, and Daniele Vigo. Bin packing approximation algorithms: survey and classification. In *Handbook of Combinatorial Optimization*, pages 455–531. Springer, 2013.

14 Edward G. Coffman Jr., Michael R. Garey, David S. Johnson, and Robert E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9(4):808–826, 1980.

15 Wenceslas Fernandez de la Vega and George S. Lueker. Bin packing can be solved within $1+\varepsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

16 Leah Epstein and Rob van Stee. This side up! *ACM Transactions on Algorithms (TALG)*, 2(2):228–243, 2006.

17 Michael R. Garey and David S. Johnson. “Strong” NP-completeness results: Motivation, examples, and implications. *JACM*, 25(3):499–508, 1978.

18 Igal Golan. Performance bounds for orthogonal oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 10(3):571–582, 1981.

19 Fabrizio Grandoni, Salvatore Ingala, and Sumedha Uniyal. Improved approximation algorithms for unsplittable flow on a path with time windows. In *WAOA*, pages 13–24, 2015.

20 Rolf Harren, Klaus Jansen, Lars Prädel, and Rob van Stee. A $(5/3+\varepsilon)$-approximation for strip packing. *Computational Geometry*, 47(2):248–267, 2014.

21 Rolf Harren and Rob van Stee. Improved absolute approximation ratios for two-dimensional packing problems. In *APPROX-RANDOM*, pages 177–189. Springer, 2009.

22 Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39–49, 2013.

23 Klaus Jansen and Lars Prädel. A new asymptotic approximation algorithm for 3-dimensional strip packing. In *SOFSEM*, pages 327–338, 2014.

24 Klaus Jansen and Roberto Solis-Oba. Rectangle packing with one-dimensional resource augmentation. *Discrete Optimization*, 6(3):310–323, 2009.

25 Klaus Jansen and Rob van Stee. On strip packing with rotations. In *STOC*, pages 755–761, 2005.

26 Klaus Jansen and Guochuan Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47(3):323–342, 2007.

27 Mohammad M. Karbasioun, Gennady Shaikhet, Evangelos Kranakis, and Ioannis Lambadaris. Power strip packing of malleable demands in smart grid. In *IEEE International Conference on Communications*, pages 4261–4265, 2013.

28 Claire Kenyon and Eric Rémila. A near-optimal solution to a two-dimensional cutting stock problem. *Math. Oper. Res.*, 25(4):645–656, 2000.

29 Arindam Khan. *Approximation Algorithms for Multidimensional Bin Packing*. PhD thesis, Georgia Institute of Technology, USA, 2015.

30 Flavio Keidi Miyazawa and Yoshiko Wakabayashi. Packing problems with orthogonal rotations. In *LATIN*, pages 359–368. Springer, 2004.

31 Giorgi Nadiradze and Andreas Wiese. On approximating strip packing better than 3/2. In *SODA*, pages 1491–1510, 2016.

32 Anshu Ranjan, Pramod Khargonekar, and Sartaj Sahni. Offline first fit scheduling in smart grids. In *IEEE SCC*, pages 758–763, 2015.

33 Ingo Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *Algorithms—ESA'94*, pages 290–299. Springer, 1994.

34 Daniel Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Information Processing Letters*, 10(1):37–40, 1980.

35 A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing*, 26(2):401–409, 1997.

36 Shaojie Tang, Qiuyuan Huang, Xiang-Yang Li, and Dapeng Wu. Smoothing the energy consumption: Peak demand reduction in smart grid. In *INFOCOM*, pages 1133–1141. IEEE, 2013.