Tight Approximation Algorithms for Two-Dimensional Guillotine Strip Packing

Arindam Khan ✉ 🏠 🆔
Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

Aditya Lonkar ✉ 🏠
Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

Arnab Maiti ✉ 🏠
Indian Institute of Technology, Kharagpur, India

Amatya Sharma ✉ 🏠
Indian Institute of Technology, Kharagpur, India

Andreas Wiese ✉ 🏠 🆔
Technische Universität München, Germany

───── **Abstract** ─────

In the STRIP PACKING problem (SP), we are given a vertical half-strip $[0, W] \times [0, \infty)$ and a set of n axis-aligned rectangles of width at most W. The goal is to find a non-overlapping packing of all rectangles into the strip such that the height of the packing is minimized. A well-studied and frequently used practical constraint is to allow only those packings that are guillotine separable, i.e., every rectangle in the packing can be obtained by recursively applying a sequence of edge-to-edge axis-parallel cuts (guillotine cuts) that do not intersect any item of the solution. In this paper, we study approximation algorithms for the GUILLOTINE STRIP PACKING problem (GSP), i.e., the STRIP PACKING problem where we require additionally that the packing needs to be guillotine separable. This problem generalizes the classical BIN PACKING problem and also makespan minimization on identical machines, and thus it is already strongly NP-hard. Moreover, due to a reduction from the PARTITION problem, it is NP-hard to obtain a polynomial-time $(3/2 - \varepsilon)$-approximation algorithm for GSP for any $\varepsilon > 0$ (exactly as STRIP PACKING). We provide a matching polynomial time $(3/2 + \varepsilon)$-approximation algorithm for GSP. Furthermore, we present a pseudo-polynomial time $(1 + \varepsilon)$-approximation algorithm for GSP. This is surprising as it is NP-hard to obtain a $(5/4 - \varepsilon)$-approximation algorithm for (general) STRIP PACKING in pseudo-polynomial time. Thus, our results essentially settle the approximability of GSP for both the polynomial and the pseudo-polynomial settings.

1   Introduction

Two-dimensional packing problems form a fundamental research area in combinatorial optimization, computational geometry, and approximation algorithms. They find numerous practical applications in logistics [9], cutting stock [23], VLSI design [26], smart-grids [20], etc. The STRIP PACKING problem (SP), a generalization of the classical BIN PACKING problem and also the makespan minimization problem on identical machines, is one of the central problems in this area. We are given an axis-aligned vertical half-strip $[0, W] \times [0, \infty)$ and a set of n axis-aligned rectangles (also called *items*) $I := \{1, 2, \ldots, n\}$, where for each rectangle i we are given an integral width $w_i \leq W$, and an integral height h_i; we assume the rectangles to be open sets. The goal is to pack all items such that the maximum height of the top edge of a packed item is minimized. The packing needs to be *non-overlapping*, i.e., such a packing into a strip of height H maps each rectangle $i \in I$ to a new translated open rectangle $R(i) := (left(i), right(i)) \times (bottom(i), top(i))$ where $right(i) = left(i) + w_i$, $top(i) = bottom(i) + h_i$, $left(i) \geq 0$, $bottom(i) \geq 0$, $right(i) \leq W$, $top(i) \leq H$ and for any $i, j \in I$, we must have $R(i) \cap R(j) = \emptyset$. We assume that items are not allowed to be rotated.

The best known polynomial time approximation algorithm for SP has an approximation ratio of $(5/3 + \varepsilon)$ for any constant $\varepsilon > 0$ [24] and a straight-forward reduction from PARTITION shows that it is NP-hard to approximate the problem with a ratio of $(3/2 - \varepsilon)$ for any $\varepsilon > 0$. Maybe surprisingly, one can approximate SP better in pseudo-polynomial time: there is a pseudo-polynomial time $(5/4 + \varepsilon)$-approximation algorithm [27] and it is NP-hard to obtain a $(5/4 - \varepsilon)$-approximation algorithm with this running time [25]. Hence, it remains open to close the gap between $(5/3 + \varepsilon)$ and $(3/2 - \varepsilon)$ for polynomial time algorithms, and even in pseudo-polynomial time, there can be no $(1 + \varepsilon)$-approximation for the problem for arbitrarily small $\varepsilon > 0$.

SP is particularly motivated from applications in which we want to cut out rectangular pieces of a sheet or stock unit of raw material, i.e., metal, glass, wood, or, cloth, and we want to minimize the amount of wasted material. For cutting out these pieces in practice, axis-parallel end-to-end cuts, called *guillotine cuts*, are popular due to their simplicity of operation [46]. In this context, we look for solutions to cut out the individual objects by a recursive application of guillotine cuts that do not intersect any item of the solution. Applications of guillotine cutting are found in crepe-rubber mills [42], glass industry [40], paper cutting [35], etc. In particular, this motivates studying geometric packing problems with the additional constraint that the placed objects need to be separable by a sequence of guillotine cuts (see Figure 1). Starting from the classical work by Christofides et al. [10] in 1970s, settings with such guillotine cuts are widely studied in the literature [16, 47, 6, 34, 15, 11, 17, 12]. In fact, many heuristics for guillotine packing have been developed to efficiently solve benchmark instances, based on tree-search, branch-and-bound, dynamic optimization, tabu search, genetic algorithms, etc. Khan et al. [32] mentions "a staggering number of recent experimental papers" on guillotine packing and lists several such recent experimental papers.

A related notion is k-stage packing, originally introduced by Gilmore and Gomory [23]. Here, each stage consists of either vertical or horizontal guillotine cuts (but not both). In each stage, each of the pieces obtained in the previous stage is considered separately and can be cut again by using either horizontal or vertical guillotine cuts. In k-stage packing, the number of cuts to obtain each rectangle from the initial packing is at most k, plus an additional cut to trim (i.e., separate the rectangles itself from a waste area). Intuitively, this means that in the cutting process we change the orientation of the cuts $k - 1$ times.
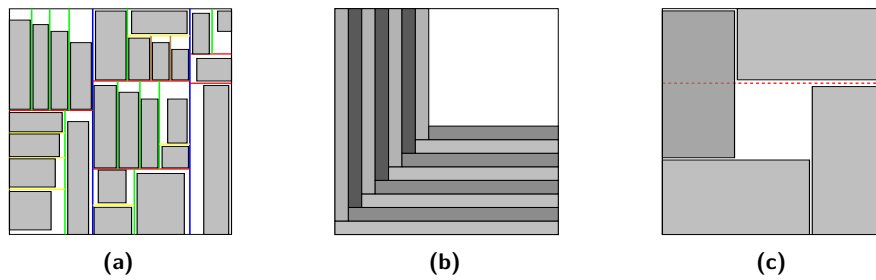
Figure 1 Packing (a) is a 5-stage guillotine separable packing, packing (b) is a $(n-1)$-stage guillotine separable packing, packing (c) is not guillotine separable as any end-to-end cut in the strip intersects a rectangle.

Therefore, in this paper, we study the GUILLOTINE STRIP PACKING problem (GSP). The input is the same as for SP, but we require additionally that the items in the solution can be separated by a sequence of guillotine cuts, and we say then that they are *guillotine separable*. Like general SP without requiring the items to be guillotine separable, GSP generalizes BIN PACKING (when all items have the same height) and makespan minimization on identical machines (when all items have the same width). Thus, it is strongly NP-hard, and the same reduction from PARTITION mentioned above yields a lower bound of $(3/2 - \varepsilon)$ for polynomial time algorithms (see the full version [31] for more details). For asymptotic approximation, GSP is well understood. Kenyon and Rémila [29] gave an asymptotic polynomial time approximation scheme (APTAS) for (general) SP. Their algorithm produces a 5-stage packing (hence, guillotine separable), and thus yields an APTAS for GSP as well. Later, Seiden et al. [43] settled the asymptotic approximation status of GSP under k-stage packing. They gave an APTAS for GSP using 4-stage guillotine cuts, and showed $k = 2$ stages cannot guarantee any bounded asymptotic performance ratio, and $k = 3$ stages lead to asymptotic performance ratios close to 1.691. However, in the non-asymptotic setting, approximation ratio of GSP is not yet settled. Steinberg's algorithm [45] yields a 2-approximation algorithm for GSP and this is the best known polynomial time approximation algorithm for the problem.

In this paper we present approximation algorithms for GSP which have strictly better approximation ratios than the best known algorithms for SP, and in the setting of pseudo-polynomial time algorithms we even beat the lower bound that holds for SP. Moreover, we show that all our approximation ratios are essentially the best possible.

1.1 Our Contribution

We present a polynomial time $(3/2 + \varepsilon)$-approximation algorithm for GSP. Due to the mentioned lower bound of $(3/2 - \varepsilon)$, our approximation ratio is essentially tight. Also, we present a pseudo-polynomial time $(1 + \varepsilon)$-approximation algorithm, which is also essentially tight since GSP is strongly NP-hard.

For the pseudo-polynomial time $(1 + \varepsilon)$-approximation, we first prove that there exists a structured solution with height at most $(1 + \varepsilon)$OPT (OPT denotes the height of the optimal solution) in which the strip is divided into $O(1)$ rectangular boxes inside which the items are *nicely packed*, e.g., horizontal items are stacked on top of each other, vertical items are placed side by side, and small items are packed greedily with the Next-Fit-Decreasing-Height algorithm [13] (see Figure 2(a) and also Figure 4. Also, refer to Section 2 for item classification). This result starkly contrasts SP (i.e., where we do not require the items to be guillotine separable): for that problem, it is already unlikely that we can prove that there
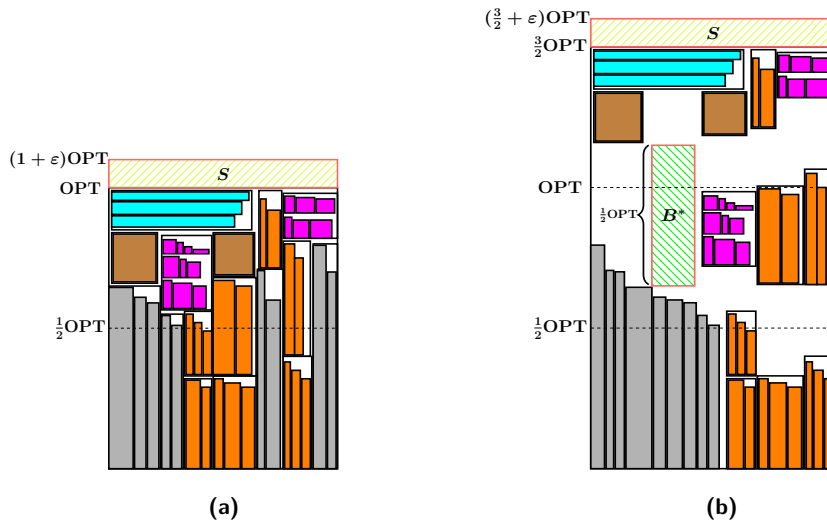
■ **Figure 2** (a) A guillotine separable structured packing (for the pseudo-polynomial time approximation scheme) where all the items are packed nicely in containers. The tall items (dark-gray) are stacked next to each other just like the vertical items (orange); the horizontal items (blue) are stacked on top of each other, the small items (pink) are packed according to NFDH, and the large containers contain single large items (brown). (b) A guillotine separable structured packing for the polynomial time $(3/2 + \varepsilon)$-approximation, where the packing from (a) is rearranged such that the tall items are *bottom-left-flushed* and there is an extra empty box B^* to accommodate some of the vertical items which we are unable to pack in polynomial time in the rest of the guessed boxes. This arises from the NP-hardness of the PARTITION problem. The yellow rectangular strip S on top of both the packings is used for packing the medium and leftover horizontal and small items.

always exists such a packing with a height of less than $5/4 \cdot \text{OPT}$. If we could prove this, we could approximate the problem in pseudo-polynomial time with a better ratio than $5/4$, which is NP-hard [25].

To construct our structured packing, we start with an optimal packing and use the techniques in [32] to obtain a packing in which each item is *nicely packed* in one of a constant number of boxes and L-shaped *compartments*. We increase the height of our packing by εOPT in order to round the heights of the packed items and get some leeway within the packing. Then, we rearrange the items placed inside the L-shaped compartments. Here, we crucially exploit that the items in the initial packing are guillotine separable. In particular, this property allows us to identify certain sets of items that we can swap, e.g., items on the left and the right of a vertical guillotine cut to simplify the packing, and reduce the number of boxes to $O(1)$. Then, using standard techniques, we compute a solution with this structure in pseudo-polynomial time and hence with a packing height of at most $(1 + \varepsilon)\text{OPT}$ (see Figure 2 (a)).

Note that we do not obtain a $(1 + \varepsilon)$-approximation algorithm in polynomial time in this way. The reason is that when we pack the items into the rectangular boxes, we need to solve a generalization of PARTITION: there can be several boxes in which vertical items are placed side by side, and we need that the widths of the items in each box sum up to at most the width of the box. If there is only a single item that we cannot place, then we would need to place it on top of the packing, which can increase our packing height by up to OPT.

For our polynomial time $(3/2 + \varepsilon)$-approximation algorithm, we, therefore, need to be particularly careful with the items whose height is larger than OPT/2, which we call the *tall items*. We prove a different structural result which is the main technical contribution of

this paper: we show that there is always a $(3/2 + \varepsilon)$-approximate packing in which the tall items are packed together in a *bottom-left-flushed* way, i.e., they are ordered non-increasingly by height and stacked next to each other with their bottom edges touching the base of the strip. All remaining items are nicely packed into $O_\varepsilon(1)$ boxes, and there is also an empty strip of height OPT/2 and width $\Omega_\varepsilon(W)$, see Figure 2 (b). Thus, it is very easy to pack the tall items correctly according to this packing. We pack the remaining items with standard techniques into the boxes. In particular, the mentioned empty strip allows us to make slight mistakes while we pack the vertical items that are not tall; without this, we would still need to solve a generalization of PARTITION.

In order to obtain our structural packing for our polynomial time $(3/2 + \varepsilon)$-approximation algorithm, we build on the idea of the packing for the pseudo-polynomial time $(1 + \varepsilon)$-approximation. Using that it is guillotine separable, we rearrange its items further. First, we move the items such that all tall items are at the bottom. To achieve this, we again argue that we can swap certain sets of items, guided by the guillotine cuts. Then, we shift certain items up by OPT/2, which leaves empty space between the shifted and the not-shifted items, see Figure 2 (b). Inside this empty space, we place the empty box of height OPT/2. Also, we use this empty space in order to be able to reorder the tall items on the bottom by their respective heights. During these changes, we ensure carefully that the resulting packing stays guillotine separable.

It is possible that also for (general) SP there always exists a structured packing of height at most $(3/2 + \varepsilon)$OPT, similar to our packing. This would yield an essentially tight polynomial time $(3/2 + \varepsilon)$-approximation for SP and thus solve the long-standing open problem to find the best possible polynomial time approximation ratio for SP. We leave this as an open question.

1.2 Other related work

In the 1980s, Baker et al. [2] initiated the study of approximation algorithms for strip packing, by giving a 3-approximation algorithm. After a sequence of improved approximations [13, 44], Steinberg [45] and Schiermeyer [41] independently gave 2-approximation algorithms. For asymptotic approximation, Kenyon and Rémila [29] settled SP by providing an APTAS.

SP has rich connections with important geometric packing problems [9, 30] such as 2D bin packing (2BP) [4, 33], 2D geometric knapsack (2GK) [19, 28], dynamic storage allocation [7], maximum independent set of rectangles (MISR) [22, 1], sliced packing [14, 18], etc.

In 2BP, we are given a set of rectangles and square bins, and the goal is to find an axis-aligned non-overlapping packing of all items into a minimum number of bins. The problem admits no APTAS [3], and the present best approximation ratio is 1.406 [4]. In 2GK, we are given a set of rectangular items and a square knapsack. Each item has an associated profit, and the goal is to pack a subset of items in the knapsack such that the profit is maximized. The present best polynomial time approximation ratio is 1.89 [19]. There is a pseudo-polynomial time $(4/3 + \varepsilon)$-approximation [21] for 2GK. In MISR, we are given a set of (possibly overlapping) rectangles we need to find the maximum cardinality non-overlapping set of rectangles. Recently, Mitchell [36] gave the first constant approximation algorithm for the problem. Then Gálvez et al. [22] obtained a $(2 + \varepsilon)$-approximation algorithm for MISR. Their algorithms are based on a recursive geometric decomposition of the plane, which can be viewed as a generalization of guillotine cuts, more precisely, to cuts with $O(1)$ bends. Pach and Tardos [38] even conjectured that for any set of n non-overlapping axis-parallel rectangles, there is a guillotine cutting sequence separating $\Omega(n)$ of them.

2BP and 2GK are also well-studied in the guillotine setting [39]. Caprara et al. [8] gave an APTAS for 2-stage SP and 2-stage BP. Later, Bansal et al. [5] showed an APTAS for guillotine 2BP. Bansal et al. [4] conjectured that the worst-case ratio between the best guillotine 2BP and the best general 2BP is 4/3. If true, this would imply a $(\frac{4}{3} + \varepsilon)$-approximation algorithm for 2BP. For guillotine 2GK, Khan et al. [32] recently gave a pseudo-polynomial time approximation scheme.

2   Pseudo-polynomial time approximation scheme

In this section, we present our pseudo-polynomial time approximation scheme (PPTAS) for GSP.

Let $\varepsilon > 0$ and assume w.l.o.g. that $1/\varepsilon \in \mathbb{N}$. We denote by OPT the height of the optimal solution. We classify the input items into a few groups according to their heights and widths similar to the classification in [32]. For two constants $1 \geq \delta > \mu > 0$ to be defined later, we classify each item $i \in I$ as:

- *tall* if $h_i > \text{OPT}/2$;
- *large* if $w_i > \delta W$ and $\text{OPT}/2 \geq h_i > \delta\text{OPT}$;
- *horizontal* if $w_i > \delta W$ and $h_i \leq \mu\text{OPT}$;
- *vertical* $w_i \leq \delta W$ and $\text{OPT}/2 \geq h_i > \delta\text{OPT}$;
- *medium if*
  - either $\delta\text{OPT} \geq h_i > \mu\text{OPT}$;
  - or $\delta W \geq w_i > \mu W$ and $h_i \leq \mu\text{OPT}$;
- *small* if $w_i \leq \mu W$ and $h_i \leq \mu\text{OPT}$;
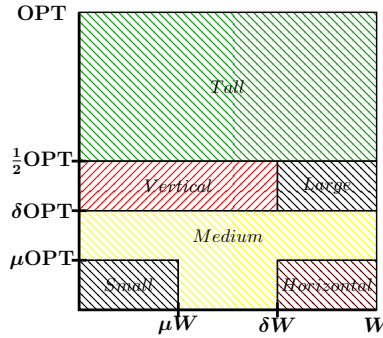
Figure 3 Item Classification: x-axis represents width and y-axis represents height.

See Figure 3 for a picture of item classification. Let $I_{tall}, I_{large}, I_{hor}, I_{ver},$ I_{medium}, I_{small} be the set of tall, large, horizontal, medium, and small rectangles in I, respectively.

Using the following lemma, one can appropriately choose μ, δ such that the medium items occupy a marginal area. This effectively allows us to ignore them in our main argumentation.

▶ **Lemma 2.1** ([37]). *Let $\varepsilon > 0$ and $f(.)$ be any positive increasing function such that $f(x) < x$ for all $x \in (0, 1]$. Then we can efficiently find $\delta, \mu \in \Omega_\varepsilon(1)$, with $\varepsilon \geq f(\varepsilon) \geq \delta \geq f(\delta) \geq \mu$ so that the total area of medium rectangles is at most $\varepsilon(\text{OPT} \cdot W)$.*

We will specify how we choose the function $f(x)$ later. In our PPTAS, we will use a packing, which is defined solely via boxes.

▶ **Definition 2.2.** *A box B is an axis-aligned open rectangle that satisfies $B \subseteq [0, W] \times [0, \infty)$. We denote by $h(B)$ and $w(B)$ the height and the width of B, respectively.*

Inside each box B, we will place the items *nicely,* meaning that they are either stacked horizontally or vertically, or B contains a single large item, or only small items, or only medium items. This is useful since in the first two cases, it is trivial to place a given set of items into B, and in the last two cases, it will turn out that it suffices to pack the items greedily using the Next Fit Decreasing Height (NFDH) algorithm [13] and Steinberg's algorithm [45], respectively. There will be one box with height at most 2εOPT that contains all medium items.

▶ **Definition 2.3** (Nice packing). *Let B be a box and let $I_B \subseteq I$ be a set of items that are placed non-overlappingly inside B. We say that the packing of I_B in B is* nice *if the items in I_B are guillotine separable and additionally*

- *I_B contains only one item, or*
- *$I_B \subseteq I_{hor}$ and the items in I_B are stacked on top of each other inside B, or*
- *$I_B \subseteq I_{tall} \cup I_{ver}$ and the items in I_B are placed side by side inside B, or*
- *$I_B \subseteq I_{medium}$, or*
- *$I_B \subseteq I_{small}$ and for each item $i \in I_B$ it holds that $w_i \leq \varepsilon \cdot w(B)$ and $h_i \leq \varepsilon \cdot h(B)$.*

We will use the term *container* to refer to a box B that contains a nice packing of some set of items I_B. See Figure 4 for nice packings in different types of containers. We say that a set of boxes $\mathcal{B}$ is *guillotine separable* if there exists a sequence of guillotine cuts that separates them and that does not intersect any box in $\mathcal{B}$.

We now state the structural lemma for the PPTAS. Intuitively, it states that there exists a $(1 + \varepsilon)$-approximate solution in which the input items are placed into $O_\varepsilon(1)$ boxes such that within each box the packing is nice. We remark that we will crucially use that in the optimal packing the items in I are guillotine separable. In fact, if one could prove that there exists such a packing with $O_\varepsilon(1)$ boxes and a height of αOPT for some $\alpha < \frac{5}{4}$ also in the non-guillotine case (where neither the optimal solution nor the computed solution needs to be guillotine separable), then one would obtain a pseudo–polynomial time $(\alpha + \varepsilon)$-approximation algorithm also in this case, by using straightforward adaptations of the algorithms in, e.g., [37, 20, 27] or our algorithm in section 2.2. However, this is not possible for $\alpha < \frac{5}{4}$, unless $\mathsf{P} = \mathsf{NP}$ [25].

▶ **Lemma 2.4** (Structural lemma 1). *Assume that μ is sufficiently small compared to δ. Then there exists a set $\mathcal{B}$ of $O_\varepsilon(1)$ pairwise non-overlapping and guillotine separable boxes all placed inside $[0, W] \times [0, (1 + 16\varepsilon)\text{OPT})$ and a partition $I = \bigcup_{B \in \mathcal{B}} I_B$ such that for each $B \in \mathcal{B}$ the items in I_B can be placed nicely into B.*

We choose our function f due to Lemma 2.1 such that μ is sufficiently small compared to δ, as required by Lemma 2.4. We will prove Lemma 2.4 in the next subsection. In its packing, let $\mathcal{B}_{hor}, \mathcal{B}_{ver}, \mathcal{B}_{tall}, \mathcal{B}_{large}, \mathcal{B}_{small}$ and $\mathcal{B}_{med}$ denote the set of boxes for the horizontal, vertical, tall, large, small and medium items, respectively. Let $\mathcal{B}_{tall+ver} := \mathcal{B}_{tall} \cup \mathcal{B}_{ver}$.

2.1 Proof of Structural Lemma 1

In this section we prove Lemma 2.4. We have omitted a few proofs which can be found in the full version [31]. Our strategy is to start with a structural lemma from [32] that guarantees the existence of a structured packing of all items in $I_{hard} := I_{tall} \cup I_{large} \cup I_{hor} \cup I_{ver}$. This packing uses boxes and **L**-compartments. Note that, for now we ignore the items I_{small}. We will show how to pack them later.
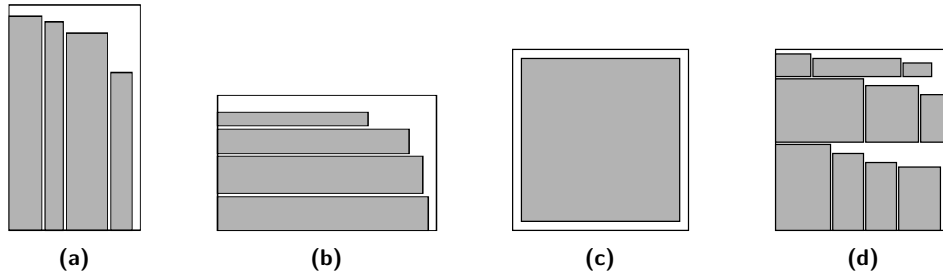
Figure 4 Nice packing of vertical, horizontal, large and small items in their respective containers.

▶ **Definition 2.5** (**L**-compartment). *An* **L**-*compartment L is an open sub-region of $[0, W] \times [0, \infty)$ bounded by a simple rectilinear polygon with six edges $e_0, e_1, \ldots, e_5$ such that for each pair of horizontal (resp. vertical) edges e_i, e_{6-i} with $i \in \{1, 2\}$ there exists a vertical (resp. horizontal) line segment ℓ_i of length less than $\delta \frac{\mathrm{OPT}}{2}$ (resp. $\delta \frac{W}{2}$) such that both e_i and e_{6-i} intersect ℓ_i but no other edges intersect ℓ_i.*

Note that for an **L**-compartment, no item $i \in I_{hor}$ can be packed in its vertical arm and similarly, no item $i \in I_{ver} \cup I_{tall}$ can be packed in its horizontal arm.

The next lemma follows immediately from a structural insight in [32] for the guillotine two-dimensional knapsack problem. It partitions the region $[0, W] \times [0, \mathrm{OPT}]$ into non-overlapping boxes and **L**-compartments that admit a *pseudo-guillotine cutting sequence*. This is a sequence of cuts in which each cut is either a (normal) guillotine cut, or a special cut that cuts out an **L**-compartment L from the current rectangular piece R in the cutting sequence, such that $R \setminus L$ is a rectangle, see Figure 6. So intuitively L lies at the boundary of R.
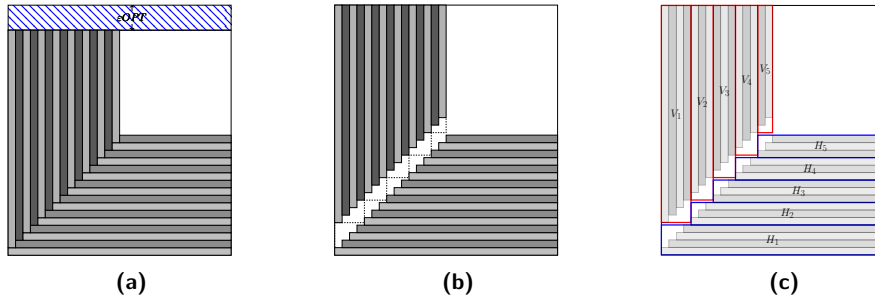
Figure 5 Using an extra $\varepsilon\mathrm{OPT}$ height, we convert a packing of items I in an **L**-compartment into another packing such that the items in I are packed in boxes $\mathcal{C}' = \mathbf{V} \cup \mathbf{H}$, which are guillotine separable and $|\mathcal{C}'| = O_\varepsilon(1)$, where $\mathbf{V} = \cup_{i=1}^{i=5} V_i$ and $\mathbf{H} = \cup_{i=1}^{i=5} H_i$.

▶ **Lemma 2.6** ([32]). *There exists a partition of $[0, W] \times [0, \mathrm{OPT}]$ into a set $\mathcal{B}_1$ of $O_\varepsilon(1)$ boxes and a set $\mathcal{L}$ of $O_\varepsilon(1)$ **L**-compartments such that*
- *the boxes and **L**-compartments in $\mathcal{B}_1 \cup \mathcal{L}$ are pairwise non-overlapping,*
- *$\mathcal{B}_1 \cup \mathcal{L}$ admits a pseudo-guillotine cutting sequence,*
- *the items in I_{hard} can be packed into $\mathcal{B}_1 \cup \mathcal{L}$ such that for each $B \in \mathcal{B}_1$ it either contains only items $i \in I_{tall} \cup I_{large} \cup I_{ver}$ or it contains only items $i \in I_{hor}$.*

Our strategy is to take the packing due to Lemma 2.6 and transform it step by step until we obtain a packing that corresponds to Lemma 2.4. First, we round the heights of the tall, large, and vertical items such that they are integral multiples of $\delta^2\mathrm{OPT}$. Formally, for each item $i \in I_{tall} \cup I_{large} \cup I_{ver}$ we round its height to $h'_i := \left\lceil \frac{h_i}{\delta^2\mathrm{OPT}} \right\rceil \delta^2\mathrm{OPT}$. Let I_{hard} denote

the resulting set of items. By a shifting argument, we will show that we can still pack I_{hard} into $O_\varepsilon(1)$ guillotine separable boxes and **L**-compartments if we can increase the height of the packing by a factor $1 + \varepsilon$ which also does not violate guillotine separability. Then, we increase the height of the packing by another factor $1 + \varepsilon$. Using this additional space, we shift the items inside each **L**-compartment L such that we can separate the vertical items from the horizontal items (see Figure 5). Due to this separation, we can partition L into $O_\varepsilon(1)$ boxes such that each box contains only horizontal or only vertical and tall items. Note however, that they might not be packed nicely inside these boxes.

▶ **Lemma 2.7.** *There exists a partition of $[0, W] \times [0, (1 + 2\varepsilon)\mathrm{OPT}]$ into a set $\mathcal{B}_2$ of $O_\varepsilon(1)$ boxes such that*
- *the boxes in $\mathcal{B}_2$ are pairwise non-overlapping and admit a guillotine cutting sequence,*
- *the items in I_{hard} can be packed into $\mathcal{B}_2$ such that they are guillotine separable and each box $B \in \mathcal{B}_2$ either contains only items from $I_{tall} \cup I_{large} \cup I_{ver}$, or contains only items from I_{hor}.*
- *Any item $i \in I_{tall} \cup I_{large} \cup I_{ver}$ has height $h_i' = k_i \delta^2 \mathrm{OPT}$ for integer k_i, $k_i \leq 1/\delta^2 + 1$.*

Let $\mathcal{B}_2$ be the set of boxes due to Lemma 2.7. Consider a box $B \in \mathcal{B}_2$ and let $I_{hard}(B)$ denote the items from I_{hard} that are placed inside B in the packing due to Lemma 2.7. Our goal is to partition B into $O_\varepsilon(1)$ smaller containers, i.e., the items in $I_{hard}(B)$ are packed nicely into these smaller boxes. If B contains horizontal items, then this can be done using standard techniques, e.g., by 1D resource augmentation (only in height) in [32]. This resource augmentation procedure maintains guillotine separability.

▶ **Lemma 2.8** ([32]). *Given a box $B \in \mathcal{B}_2$ such that B contains a set of items $I_{hard}(B) \subseteq I_{hor}$. There exists a partition of B into $O_{\varepsilon'}(1)$ containers $\mathcal{B}'$ and one additional box B' of height at most $\varepsilon' h(B)$ and width $w(B)$ such that the containers $\mathcal{B}'$ are guillotine separable and the containers $\mathcal{B}' \cup \{B'\}$ contain $I_{hard}(B)$.*

We apply Lemma 2.8 to each box $B \in \mathcal{B}_2$ that contains a horizontal item. Consider the items which are contained in their respective boxes B'. In order to avoid any confusions between constants of our algorithm and resource augmentation, we denote the constant used for resource augmentation as ε'. We choose $\varepsilon' = \varepsilon$ and then their total area is at most $\varepsilon \mathrm{OPT} \cdot W$ and therefore, all such items can be packed in a box of height at most $2\varepsilon \mathrm{OPT}$ and width W using Steinberg's algorithm [45]. But since this will possibly not result in a nice packing we apply resource augmentation (only along height) again to ensure that we get a nice packing of such horizontal items in $O_\varepsilon(1)$ containers which can all be packed in a box of height at most $3\varepsilon \mathrm{OPT}$ and width W.

Consider now a box $B \in \mathcal{B}_2$ that contains at least one item from $I_{tall} \cup I_{large} \cup I_{ver}$. Let $I_{hard}(B) \subseteq I_{tall} \cup I_{large} \cup I_{ver}$ denote the items packed inside B. We argue that we can rearrange the items in $I_{hard}(B)$ such that they are nicely placed inside $O_\varepsilon(1)$ containers. In this step we crucially use that the items in $I_{hard}(B)$ are guillotine separable.

Consider the guillotine cutting sequence for $I_{hard}(B)$. It is useful to think of these cuts as being organized in *stages:* in the first stage we do vertical cuts (possibly zero cuts). In the following stage, we take each resulting piece and apply horizontal cuts. In the next stage, we again take each resulting piece and apply vertical cuts, and so on. Since the heights of the items in $I_{hard}(B)$ are rounded to multiples of $\delta^2 \mathrm{OPT}$ we can assume w.l.o.g. that the y-coordinates of the horizontal cuts are all integral multiples of $\delta^2 \mathrm{OPT}$ (possibly moving the items a little bit). Assume here for the sake of simplicity that $t = 1/\delta^2$ is an integer. Because of the rounding of heights of the items in $I_{hard}'(B)$, there are at most $(1/\delta^2 - 1)$ y-coordinates for making a horizontal cut. For a horizontal stage of cuts, for a rectangular

(a)   **(b)**
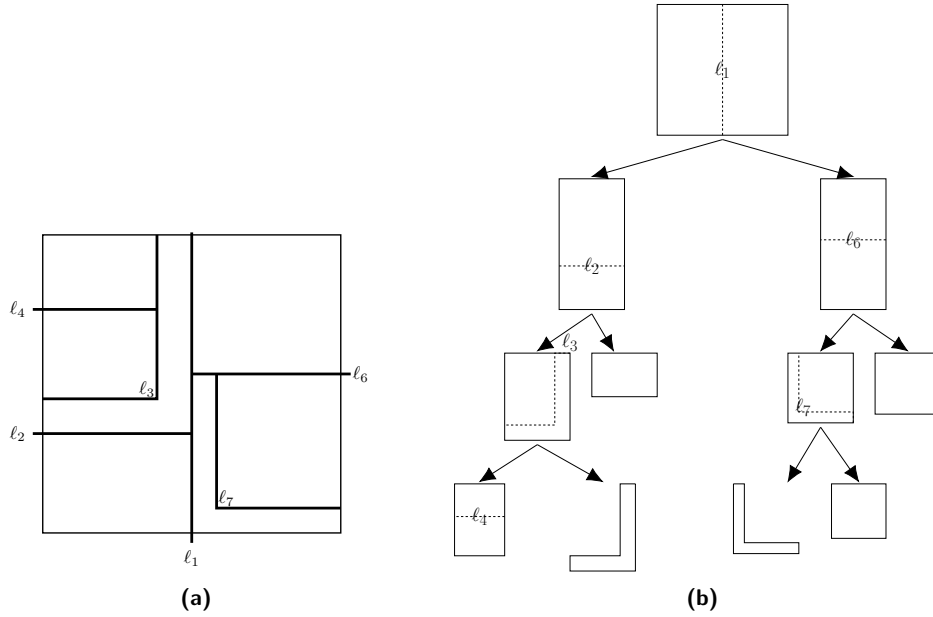
Figure 6 (a) A pseudo-guillotine cutting sequence. The first cut is l_1, and then the resulting left piece is further subdivided by ℓ_2, ℓ_3 and ℓ_4. Similarly, ℓ_6, ℓ_7 subdivide the right piece. Note that ℓ_3 and ℓ_7 are not guillotine cuts, but they cut out the corresponding **L**-compartments. (b) Step by step pseudo-guillotine cutting sequence corresponding to Figure (a). Dashed line at each level indicates a partition of a rectangle into two regions (two boxes, or one box and one **L**).

piece we define a *configuration vector* $(x_1, ..., x_{t-1})$: For each $i \in [t-1]$ if there is a horizontal cut in the piece at $y = t \cdot i$, then $x_i = 1$, otherwise $x_i = 0$. Consider $y = 0$ to be the bottom of the rectangular piece. Therefore, in each horizontal stage, for each piece there are at most $K := (2^{(1/\delta^2)})$ possible configurations. Consider the first stage (which has vertical cuts). If there are more than K vertical cuts then in two of the resulting pieces, in the second stage the same configuration of horizontal cuts is applied (see Figure 7).

We reorder the resulting pieces and their items such that pieces with the same configuration of horizontal cuts are placed consecutively. Therefore, in the first stage we need only K vertical cuts and we can have at most $(\frac{1}{\delta} 2^{(1/\delta^2)})$ resulting pieces. We apply the same transformation to each stage with vertical cuts. Now observe that there can be at most $O(1/\delta)$ stages since there are at most $1/\delta$ possible tall, vertical or large items stacked on top of the other and thus at most $1/\delta$ stages with horizontal cuts. Therefore, after our transformations, we apply only $(\frac{1}{\delta} 2^{(1/\delta^2)})^{\frac{1}{\delta}}$ cuts in total, in all stages in all resulting pieces. Thus, we obtain $O_\varepsilon(1)$ boxes at the end, in which the items are nicely packed. This leads to the following lemma.

▶ **Lemma 2.9.** *Given a box $B \in \mathcal{B}_2$ such that B contains a set of items $I_{hard}(B) \subseteq I_{tall} \cup I_{large} \cup I_{ver}$. There exists a partition of B into $O_\varepsilon(1)$ containers $\mathcal{B}'$ such that the containers $\mathcal{B}'$ are guillotine separable and contain the items $I_{hard}(B)$.*

We apply Lemma 2.9 to each box $B \in \mathcal{B}_2$ that contains an item from $I_{tall} \cup I_{large} \cup I_{ver}$. Thus, we obtain a packing of I_{hard} into a set of $O_\varepsilon(1)$ guillotine separable containers in which these items are nicely placed; we denote these containers by $\mathcal{B}_{hard}$. This yields directly a packing for the (original) items I_{hard} (without rounding). Finally, we partition the empty space of the resulting packing into more boxes, and one additional box that we place on top
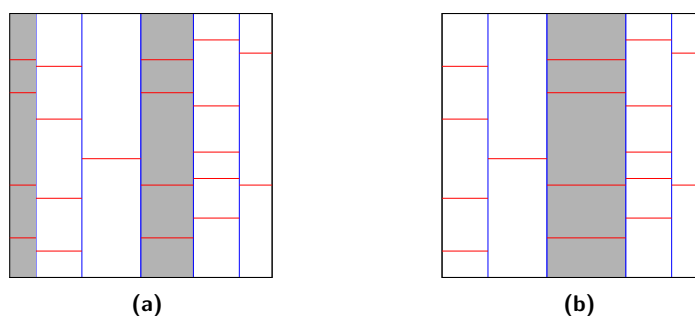
Figure 7 (a) 2 stages of guillotine cuts for a box containing vertical rectangles. (b) Since rounded heights of vertical rectangles are integral multiples of δ^2, merge configurations with same set of horizontal cuts to get $O_\delta(1)$ configurations.

of the current packing. We pack the items in I_{small} inside all these boxes. We might not be able to use some parts of the empty space, e.g., if two boxes are closer than μW to each other horizontally; however, if μ is sufficiently small compared to the number of boxes, this space is small and compensated by the additional box.

▶ **Lemma 2.10.** *Assume that μ is sufficiently small compared to δ. There exists a set of $O_\varepsilon(1)$ boxes $\mathcal{B}_{small}$, all contained in $[0, W] \times [0, (1 + 14\varepsilon)\mathrm{OPT}]$, such that the boxes in $\mathcal{B}_{hard} \cup \mathcal{B}_{small}$ are non-overlapping and guillotine separable and the items in I_{small} can be placed nicely into the boxes $\mathcal{B}_{small}$.*

Finally, we show the following lemma by using the fact that the medium items have area at most $\varepsilon(\mathrm{OPT} \cdot W)$ and by applying Steinberg's algorithm [45]. This completes the proof of Lemma 2.4.

▶ **Lemma 2.11.** *In time $n^{O(1)}$ we can find a nice placement of all items in I_{medium} inside one container B_{med} of height $2\varepsilon\mathrm{OPT}$ and width W.*

2.2 Algorithm

We describe now our algorithm that computes a packing of height at most $(1 + O(\varepsilon))\mathrm{OPT}$. First, we guess OPT and observe that there are at most $n \cdot h_{\max}$ possibilities, where $h_{\max} := \max_{i \in I} h_i$. Then, we guess the set of containers $\mathcal{B}$ due to Lemma 2.4 and their placement inside $[0, W] \times [0, (1 + O(\varepsilon))\mathrm{OPT}]$. For each container $B \in \mathcal{B}$ we guess which case of Definition 2.3 applies to B, i.e., whether I_B contains only one item, $I_B \subseteq I_{hor}$, $I_B \subseteq I_{tall} \cup I_{ver}$, $I_B \subseteq I_{medium}$, or $I_B \subseteq I_{small}$. For each box $B \in \mathcal{B}$ for which I_B contains only one item $i \in I$, we guess i. Observe that for the remaining containers this yields independent subproblems for the sets I_{hor}, $I_{tall} \cup I_{ver}$, I_{medium}, and I_{small}. We solve these subproblems via similar routines as in [37, 20, 27].

We pack all medium items in I_{medium} into one single container B_{med} of height $2\varepsilon\mathrm{OPT}$ by Lemma 2.11. Then, for the sets I_{hor} and $I_{tall} \cup I_{ver}$ we pack their respective items into their containers using a standard pseudo-polynomial time dynamic program; we denote these containers by $\mathcal{B}_{hor}$ and $\mathcal{B}_{tall+ver}$, respectively. We crucially use that $|\mathcal{B}_{hor}| \leq O_\varepsilon(1)$ and $|\mathcal{B}_{tall+ver}| \leq O_\varepsilon(1)$. See the full version [31] for the details of packing of items in I_{hor} and $I_{tall} \cup I_{ver}$.

Finally, we pack the small items. From the proof of Lemma 2.10, apart from some items $I'_{small} \subset I_{small}$ which have area at most $\varepsilon\mathrm{OPT} \cdot W$, the other items can be packed nicely in the containers in $\mathcal{B}_{small} \setminus B_{small}$, where B_{small} has height $9\varepsilon\mathrm{OPT}$ and width W. Thus,

we use NFDH for packing the remaining small items. It can be shown that the small items which remain unpacked can be packed nicely in B_{small}, which is placed on the top of our packing.

▶ **Theorem 2.12.** *There is a $(1 + \varepsilon)$-approximation algorithm for the guillotine strip packing problem with a running time of $(nW)^{O_\varepsilon(1)}$.*

3 Polynomial time $(\frac{3}{2} + \varepsilon)$-approximation

In this section, we first present the structural lemma for our polynomial time $(3/2 + \varepsilon)$-approximation algorithm for guillotine strip packing. Then we describe our algorithm. We have omitted a few proofs which can be found in the full version [31].

To derive our structural lemma, we start with the packing due to Lemma 2.4. The problem is that with a polynomial time algorithm (rather than a pseudo-polynomial time algorithm) we might not be able to pack all tall items in their respective boxes. If there is even one single tall item i that we cannot pack, then we need to place i on top of our packing, which can increase the height of the packing by up to OPT.

(a)   **(b)**   **(c)**   **(d)**
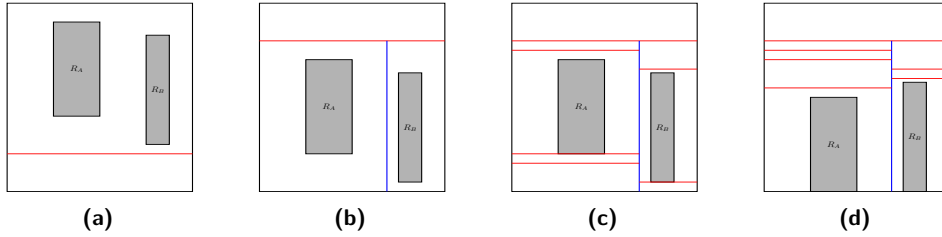
Figure 8 R_A and R_B are tall containers and by swapping the respective boxes (forming as a results of guillotine cuts) that contain them, they can be packed such that the bottoms of both containers intersect the bottom of the strip.

Therefore, we make our packing more robust to small errors when we pack the items into their boxes. In our changed packing, the tall items are *bottom-left-flushed* (see Figure 9(f)), the remaining items are packed into $O_\varepsilon(1)$ boxes, and there is one extra box B^* of height OPT/2 and width $\Omega_\varepsilon(W)$ which is empty. We will use the extra box B^* in order to compensate small errors when we pack the vertical items.

Formally, we say that in a packing, a set of items I' is *bottom-left-flushed* if they are ordered non-increasingly by height and stacked next to each other in this order within the strip $[0, W] \times [0, \infty)$ starting at the left edge of the strip, such that the bottom edge of each item $i \in I'$ touches the line segment $[0, W] \times \{0\}$. We now state the modified structural lemma for our polynomial time $(3/2 + \varepsilon)$-approximation algorithm formally.

▶ **Lemma 3.1** (Structural lemma 2). *There exists a packing of the items I within $[0, W] \times [0, (3/2 + O(\varepsilon))\text{OPT})$ such that*
- *The items I_{tall} are bottom-left-flushed,*
- *There is a set $\mathcal{B}$ of $O_\varepsilon(1)$ containers that are pairwise non-overlapping and do not intersect the items in I_{tall},*
- *There is a partition of $I \setminus I_{tall} = \bigcup_{B \in \mathcal{B}} I_B$ such that for each $B \in \mathcal{B}$ the items in I_B can be placed nicely into B,*
- *There is a container $B^* \in \mathcal{B}$ of height OPT/2 and width $\varepsilon_1 W$ such that $I_{B^*} = \emptyset$,*
- *The items I_{tall} and the containers $\mathcal{B}$ together are guillotine separable.*

We now prove Lemma 3.1 in the following subsection.

3.1   Proof of Structural Lemma 2

We start with the packing due to Lemma 2.4 and transform it step by step. To obtain our packing, we first argue that we can ensure that all tall items are placed on the bottom of the strip, i.e., their bottom edges touch the bottom edge of the strip. Here we use that the initial packing is guillotine separable. Then we place the box B^* as follows. Suppose that there are initially C containers that cross the horizontal line with $y = \text{OPT}/2$. Note that $C = O_\varepsilon(1)$ and $C \geq 1$ since at least one container is required to pack given non-zero number of items. Then, by an averaging argument we can show that there is a line segment l^* of length at least $\Omega(\frac{W}{C})$ which is the top edge of one of the containers B in the packing at some height $h^* \geq \text{OPT}/2$. We push all the containers which completely lie above the line $y = h^*$ vertically upward by $\text{OPT}/2$ and this creates enough space to pack B^* on top of B. After that, we take advantage of the gained extra space in order to ensure that the tall items are bottom-left-flushed.

Now we describe the proof formally. First, we define some constants. Let $g(\delta, \varepsilon) = O_\varepsilon(1)$ denote an upper bound on the number of containers in the packing obtained using Lemma 2.4, depending on ε and δ. Let $\varepsilon_1 = \frac{1}{3g(\delta,\varepsilon)}$, $\varepsilon_2 = \frac{\varepsilon}{4|\mathcal{B}_{hor}|}$, $\varepsilon_3 = \frac{\varepsilon_1}{4|\mathcal{B}_{ver}|}$, $\varepsilon_4 = \mu$, $\varepsilon_5 = \frac{\varepsilon_1 \delta}{6}$, $\varepsilon_6 = \frac{\varepsilon \delta}{6}$.

Our first goal is to make sure that the tall items are all placed on the bottom of the strip $[0, W] \times [0, \infty)$. For this, we observe the following: suppose that in the guillotine cutting sequence a horizontal cut is placed. This cut separates the current rectangular piece R into two smaller pieces R_1 and R_2. Suppose that R_1 lies on top of R_2. Then only one of the two pieces R_1, R_2 can contain a tall item. Also, we obtain an alternative guillotine separable packing if we swap R_1 and R_2 – together with the items contained in them – within R. We perform this swap if R_1 contains a tall item. We apply this operation to each horizontal cut in the guillotine cutting sequence. As a result, we obtain a new packing in which all tall items are placed on the bottom of the strip (but possibly not yet bottom-left-flushed) as shown in Fig 8.

▶ **Lemma 3.2.** *There exists a set $\mathcal{B}$ of $O_\varepsilon(1)$ pairwise non-overlapping and guillotine separable boxes that are all placed inside $[0, W] \times [0, (1 + 16\varepsilon)\text{OPT})$ and a partition $I = \bigcup_{B \in \mathcal{B}} I_B$ such that for each $B \in \mathcal{B}$ the items in I_B can be placed nicely into B. Also, for each box $B \in \mathcal{B}$ with $I_B \cap I_{tall} \neq \emptyset$ we have that the bottom edge of B intersects the line segment $[0, W] \times \{0\}$.*

Let $\mathcal{B}$ be the set of containers due to Lemma 3.2. We want to move some of them up in order to make space for the additional box B^*. To this end, we identify a horizontal line segment ℓ^* in the following lemma.

▶ **Lemma 3.3.** *There is a horizontal line segment ℓ^* of width at least $\varepsilon_1 W$ that does not intersect any container in $\mathcal{B}$, and such that the y-coordinate of ℓ^* is at least $\text{OPT}/2$.*

Proof. Consider the containers in $\mathcal{B}$ that intersect with the horizontal line segment $\ell := [0, W] \times \{\text{OPT}/2\}$ and let $p_1, ... p_k$ be the maximally long line segments on ℓ that do not intersect any container. Since the line segments $\{p_1, ... p_k\}$ are between containers in $\mathcal{B}$, we have that $k \leq |\mathcal{B}| + 1$. Therefore by an averaging argument we can find a horizontal line segment ℓ^* of width at least $\frac{W}{2(g(\delta,\varepsilon))+1} \geq \frac{W}{3g(\delta,\varepsilon)} \geq \varepsilon_1 W$ that either contains the top edge of one of these containers such that ℓ^* does not intersect any other container in $\mathcal{B}$ or ℓ^* is one of the line segments in the set $\{p_1, ..., p_k\}$. Hence, the y-coordinate of ℓ^* is at least $\text{OPT}/2$.   ◄

Let h^* be the y-coordinate of ℓ^*. We take all containers in $\mathcal{B}$ that lie "above h^*", i.e., that lie inside $[0, W] \times [h^*, \infty)$. We translate them up by $\text{OPT}/2$. We define a container B^* which has height $\text{OPT}/2$ and width $\varepsilon_1 W$ to be packed such that ℓ^* is the bottom edge of B^* (see Figure 9(b)). We then make the following claim about the resulting packing of $\mathcal{B} \cup \{B^*\}$ (we call this packing P_1).

(a) **(b)** **(c)**
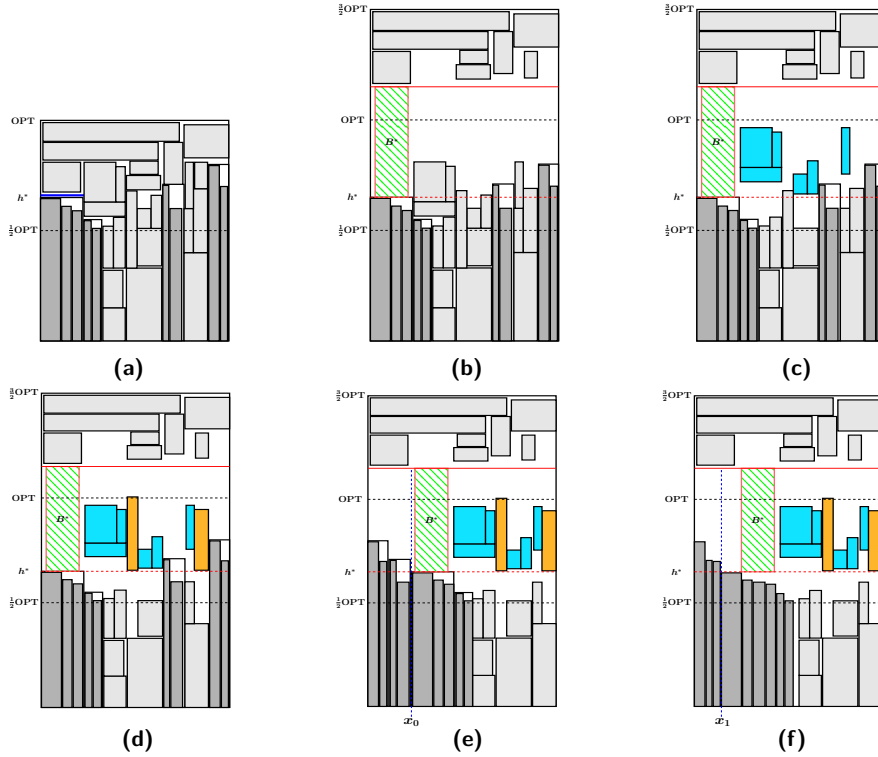
(d) **(e)** **(f)**

■ **Figure 9** (a) A guillotine separable packing with items nicely packed in containers. The gray colored rectangles are the tall items and the light-gray rectangles are containers with items nicely packed inside. The blue line segment indicates l^* at height $h^\star$. (b) Items completely packed in $[h^*, \mathrm{OPT}]$ are shifted by $\frac{1}{2}\mathrm{OPT}$ vertically upward. The thick red line indicates $y = h^* + \frac{1}{2}\mathrm{OPT}$ which separates the items shifted up from the items below. The dashed red line indicates the height h^* and B^* is packed in the strip of sufficient width and lowest height h^*. (c) The containers of type 1 (colored blue) are moved accordingly so they do not intersect $y = h^\star$. (d) The containers of type 2 (colored yellow) are moved accordingly so they do not intersect $y = h^\star$. (e) The containers in $\mathcal{B}_{tall}$ are *bottom-left-flushed* while other non-tall containers are moved accordingly to the right. The blue vertical dashed line $x = x_0$ separates containers in $\mathcal{B}_{tall}^+$ to its left hand side from other containers to the right. (f) Final packing where tall items are *bottom-left-flushed* and the blue vertical dashed line $x = x_1$ separates items $i \in I_{tall}$ with $h_i > h^*$ to the left from other items and containers to the right.

▶ **Lemma 3.4.** *The packing P_1 is feasible, guillotine separable and has height $(3/2 + O(\varepsilon))\mathrm{OPT}$.*

Proof. Since $h^* > \mathrm{OPT}/2$, observe that no containers are intersecting the line $[0, W] \times \{h^* + \mathrm{OPT}/2\}$. This is because any containers which were lying above the line $[0, W] \times \{h^*\}$ before were pushed up by $\mathrm{OPT}/2$ and the height of such containers is at most $\mathrm{OPT}/2$. Thus, the first guillotine cut is applied at $y = h^* + \mathrm{OPT}/2$ so that we get two pieces R and R_{top}. For the guillotine separability of the top piece R_{top}, we use the fact that the packing to begin with was guillotine separable and we have moved a subset of the items in the initial packing vertically upwards by the same height. For the bottom piece R, which has a subset of the initial packing, we have packed B^* on the top edge (which is part of the line $[0, W] \times \{h^*\}$) of another container (say B) whose width is more than the width of B^*. In the guillotine cutting sequence of this piece without the addition of B^*, consider the horizontal cuts at height at least h^*. Note that there is no container lying completely above the line $[0, W] \times \{h^*\}$ in R.

Hence, we can remove such horizontal cuts and extend the vertical cuts that were intercepted by these horizontal cuts until they hit the topmost horizontal edge of R. Now, if we follow this new guillotine cutting sequence, we would finally have a rectangular region with only the container B. As there is no container in the region $[left(B), right(B)] \times [h^*, h^* + \mathrm{OPT}/2]$, we can pack B^* in this region without violating the guillotine separability condition. Now, observe that the height of the piece R_{top} is $(1 + O(\varepsilon))\mathrm{OPT} - h^*$ and height of the piece R is $h^* + \mathrm{OPT}/2$. Hence the height of the packing P_1 is $(3/2 + O(\varepsilon))\mathrm{OPT}$. ◄

Our next goal is to rearrange the tall items and their containers such that the tall items are bottom-left flushed. Let $\mathcal{B}_{tall} \subseteq \mathcal{B}$ denote the containers in $\mathcal{B}$ that contain at least one tall item. Consider the line segment $\ell := [0, W] \times \{h^*\}$ and observe that it might be intersected by containers in $\mathcal{B}_{tall}$. Let $\ell_1, \ell_2, ..., \ell_t$ be the connected components of $\ell \setminus \bigcup_{B \in \mathcal{B}_{tall}} B$. For each $j \in \{1, ..., t\}$ we do the following. Consider the containers in $\mathcal{B} \setminus \mathcal{B}_{tall}$ whose bottoms are contained in $\ell_j \times [\mathrm{OPT}/2, h^*]$ (we call them *type 1 containers*). We move them up by $h^* - \mathrm{OPT}/2$ units. There is enough space for them since the top edge of any of these containers lies below the line segment $[0, W] \times \{h^* + \mathrm{OPT}/2\}$ after shifting.

Then we take all containers in $\mathcal{B} \setminus \mathcal{B}_{tall}$ that intersect ℓ_j and also the line segment $[0, W] \times \{\mathrm{OPT}/2\}$ (*type 2 containers*). We move them up such that their respective bottom edges are contained in ℓ_j. Again there is enough space for this since the containers have height at most $\mathrm{OPT}/2$ and hence, their top edges cannot cross the line segment $[0, W] \times \{h^* + \mathrm{OPT}/2\}$. Note that in this step we do not necessarily move the affected containers uniformly. See Figure 9(c) and Figure 9(d) for a sketch. Note that due to the way ℓ^* is defined, no type 1 or type 2 container after being shifted overlaps with the region occupied by B^*.

One can show that the resulting packing is still guillotine separable. In particular, there is such a sequence that starts as follows: the first cut of this sequence is a horizontal cut with y-coordinate $h^* + \mathrm{OPT}/2$. For the resulting bottom piece R, there are vertical cuts that cut through the vertical edges of the containers in $\mathcal{B}_{tall}$ whose height is strictly greater than h^*, denote these containers by $\mathcal{B}_{tall}^+$. Let $R_1, ..., R_{t'}$ denote the resulting partition of R. We can rearrange our packing by reordering the pieces $R_1, ..., R_{t'}$. We reorder them such that on the left we place the pieces containing one container from $\mathcal{B}_{tall}^+$ each, sorted non-increasingly by their heights. Then we place the remaining pieces from $\{R_1, ..., R_{t'}\}$ (which hence, do not contain any containers in $\mathcal{B}_{tall}^+$), denote their union by R'. Let the left end of R' be $x = x_0$. We can assume that the guillotine cutting sequence places a vertical cut that separates R' from the other pieces in $\{R_1, ..., R_{t'}\}$ at $x = x_0$. From Lemma 3.3, we know that there is a container B (or possibly the case when $h^* = \mathrm{OPT}/2$ and we have a line segment ℓ' of width at least $\varepsilon_1 W$ on top of which we can pack B^*) whose top is at height h^*, has width at least ℓ^* which now lies to the right of x_0 in R'. Thus, the region $[left(B), left(B) + \varepsilon_1 W] \times [h^*, h^* + \mathrm{OPT}/2]$ is empty and can be used to place B^*.

We change now the placement of the containers within R'. Due to our rearrangements, no container inside R' intersects the line segment $[0, W] \times \{h^*\}$, so we can assume that R' is cut by the horizontal cut $[0, W] \times \{h^*\}$, let R'' be the resulting bottom piece and R''' be the piece above. We first show why R''' is guillotine separable. First, we separate B^* using vertical guillotine cuts at its left and right edges. Then we prove that the shifting operation for type 2 and type 1 containers does not violate guillotine separability of the packing for any region defined by some horizontal segment l_j for $j \in [t]$. Consider any type 2 container B'. Its top edge was initially lying above $y = h^*$ and its bottom below $\mathrm{OPT}/2$. Hence, before shifting this container no item could have been packed such that it was in the region $[0, W] \times [h^*, h^* + \mathrm{OPT}/2]$ and was intersecting the vertically extended line segments from the left and right edges of B' because any item packed in $[0, W] \times [h^*, \infty)$ initially was shifted

upward by OPT/2. Hence, after shifting B' such that its bottom touches $y = h^*$, after considering the cut $y = h^*$ in l_j, extend its left and right edges vertically upward to separate B' using guillotine cuts. For the type 1 containers, after the aforementioned cuts observe that all such containers have been shifted by an equal amount vertically upward and using the fact that they were guillotine separable initially, we claim that they are guillotine separable afterward. This is proved by considering the initial guillotine cuts that were separating such items and shifting the horizontal cuts upward by $h^* - \text{OPT}/2$ (equal to the distance the type 1 containers were shifted upward by).

To show that R'' is guillotine separable, observe that due to our rearrangements there are no containers that are completely contained in $R'' \cap ([0, W] \times [\text{OPT}/2, h^*])$. Therefore, we can assume that the next cuts for R'' are vertical cuts that contain all vertical edges of the boxes in $\mathcal{B}_{tall}$ that are contained in R''. Let $R_1'', ..., R_{t''}''$ denote the resulting pieces. Like above, we change our packing such that we reorder the pieces in $R_1'', ..., R_{t''}''$ non-increasingly by the height of the respective box in $\mathcal{B}_{tall}$ contained in them, and at the very right we place the pieces from $R_1'', ..., R_{t''}''$ that do not contain any container from $\mathcal{B}_{tall}$ (see Figure 9(e))

Finally, we sort the tall items inside the area $\bigcup_{B \in \mathcal{B}_{tall}} B$ non-increasingly by height so that they are bottom-left-flushed, and we remove the containers $\mathcal{B}_{tall}$ from $\mathcal{B}$ (see Figure 9(f)). We now prove that the tall items can be sorted inside the area $\bigcup_{B \in \mathcal{B}_{tall}} B$ non-increasingly by height without violating guillotine separability and feasibility. Note that the area $\bigcup_{B \in \mathcal{B}_{tall}} B$ can possibly contain some vertical items. Now, we reorder the tall items within R' such that they are sorted in non-increasing order of their heights. We do the same for all the tall items on the left of R'. There may be tall items (or vertical items) on the left hand side of R' such that for any such item, its height is less than the the tallest tall item in R'. Note that such tall items have to have a height of at most h^*. Such items can be repeatedly swapped with their neighboring tall item till they are in the correct position according to the *bottom-left-flushed* packing of the tall items, while maintaining guillotine separability. Such a swap operation between consecutive tall items ensures that all of the tall items and possibly some vertical items which were initially packed in tall containers remain inside the area $\bigcup_{B \in \mathcal{B}_{tall}} B$. We ensure that the vertical items which were packed to the left of R' get swapped so that they are packed on the right of all the tall items in a container. Now, to prove that guillotine separability of the packing is maintained after all such swapping operations, that is, after all tall items are sorted according to their heights in a non-increasing order consider the x-coordinate (say x_1) of the right edge of the shortest tall item which has height strictly greater than h^*. Observe that there were no tall containers of height strictly greater than h^* beyond $x = x_0$, which implies $x_1 \leq x_0$ and hence, now, for the guillotine cutting sequence, we can have a vertical guillotine cut at $x = x_1$ instead of at $x = x_0$, the rest being the same as mentioned before. This yields the packing claimed by Lemma 3.1.

3.2 Algorithm for polynomial time $(\frac{3}{2} + \varepsilon)$-approximation

First we guess a value OPT' such that $\text{OPT} \leq \text{OPT}' \leq (1 + \varepsilon)\text{OPT}$ in $n^{O_\varepsilon(1)}$ time (see the full version [31] for the details). In order to keep the notation light we denote OPT' by OPT. We want to compute a packing of height at most $(\frac{3}{2} + O(\varepsilon))\text{OPT}$ using Lemma 3.1.

Intuitively, we first place the tall items in a bottom-left-flushed way. Then we guess approximately the sizes of the boxes, place them in the free area, and place the items inside them via guessing the relatively large items, solving an instance of the generalized assignment problem (GAP), using NFDH for the small items, and invoking again Lemma 2.11 for the medium items. This is similar as in, e.g., [19, 28].

Formally, first we place all items in I_{tall} inside $[0, W] \times [0, (3/2 + \varepsilon)\text{OPT})$ such that they are bottom-left-flushed. Then, we guess approximately the sizes of the containers in $\mathcal{B}$. Note that in polynomial time we cannot guess the sizes of the containers exactly. Let $B \in \mathcal{B}$. Depending on the items packed inside B, we guess different quantities for B.

- If there is only one single large item $i \in I$ packed inside B then we guess i.
- If B contains only items from I_{hor} then we guess the widest item packed inside B. This defines our guessed width of B. Also, we guess all items packed inside B whose height is at least $\varepsilon_2\text{OPT}$ (at most $O(1/\varepsilon_2)$ many), denote them by I'_B. We guess the total height of the remaining items $I_B \setminus I'_B$ approximately by guessing the quantity $\hat{h}(B) := \left\lfloor \frac{h(I_B \setminus I'_B)}{\varepsilon_2\text{OPT}} \right\rfloor \varepsilon_2\text{OPT}$. Our guessed height for B is then $\sum_{i \in I'_B} h(i) + \hat{h}(B)$.
- Similarly, if B contains only items from I_{ver} then we guess the highest item packed inside B, which defines our guessed height of B. Also, we guess all items packed inside B whose width is at least $\varepsilon_3 W$ (at most $O(1/\varepsilon_3)$ many), denote them by I'_B. We guess the total width of the remaining items $I_B \setminus I'_B$ approximately by guessing the quantity $\hat{w}(B) := \left\lfloor \frac{w(I_B \setminus I'_B)}{\varepsilon_3\text{OPT}} \right\rfloor \varepsilon_3\text{OPT}$ and our guessed width of B is then $\sum_{i \in I'_B} w(i) + \hat{w}(B)$.
- If B contains only small items, then our guessed heights and widths of B are $\left\lfloor \frac{h(B)}{\varepsilon_4\text{OPT}} \right\rfloor \varepsilon_4\text{OPT}$ and $\left\lfloor \frac{w(B)}{\varepsilon_4 W} \right\rfloor \varepsilon_4 W$, respectively.

Note that here $\varepsilon_2 = \frac{\varepsilon}{4|\mathcal{B}_{hor}|}$, $\varepsilon_3 = \frac{\varepsilon_1}{4|\mathcal{B}_{ver}|}$ and $\varepsilon_4 = \mu$ are chosen so that the unpacked horizontal items, unpacked vertical items and unpacked small items due to container rounding can be packed in containers B_{hor} (defined below), B^* and B_{small}, respectively.

We have at most $O_\varepsilon(1)$ containers and for each container $B \in \mathcal{B}$ we guess the type of container B and its respective width and height (depending on the type) in $n^{O_\varepsilon(1)}$ time.

Additionally, we guess three containers B_{med} of height $2\varepsilon\text{OPT}$, B_{hor} of height εOPT, and B_{small} of height $27\varepsilon\text{OPT}$ and width W each that we will use to place all medium items, and to compensate errors due to inaccuracies of our guesses for the sizes of the containers for horizontal and small items, respectively. Let $\mathcal{B}'$ denote the guessed containers (including B_{med}, B_{hor}, and B_{small}). Since $|\mathcal{B}'| = O_\varepsilon(1)$ and the containers in $\mathcal{B}'$ are not larger than the containers in $\mathcal{B}$, we can guess a placement for the containers $\mathcal{B}'$ such that together with I_{tall} they are guillotine separable. We place the containers B_{med}, B_{hor}, and B_{small} on top of the packing of rest of the containers in $\mathcal{B}'$, and I_{tall}.

▶ **Lemma 3.5.** *In time $n^{O_\varepsilon(1)}$ we can compute a placement for the containers in $\mathcal{B}'$ such that together with the items I_{tall}, they are guillotine separable.*

Next, we place the vertical items. Recall that for each container $B \in \mathcal{B}$ containing items from I_{ver} we guessed the items packed inside B whose width is at least $\varepsilon_3 W$. For each such container B we pack these items into the container $B' \in \mathcal{B}'$ that corresponds to B. With a similar technique as used for the generalized assignment problem (GAP) [19], we place all but items with width at most $\varepsilon_3 W$ for each container in I_{ver}. Further using the PTAS for this variant of GAP, we can ensure that items of total area at most $3\varepsilon_5 \cdot \text{OPT} \cdot W$ are not packed. Hence, items of total width at most $(3\varepsilon_5/\delta)W$ remain unpacked as each such item has height at least δOPT. We pack these remaining items into B^*, using that each of them has a height of at most $\text{OPT}/2$ and that their total width is at most $|\mathcal{B}'| \cdot 2\varepsilon_3 W + (3\varepsilon_5/\delta)W \leq \varepsilon_1 W = w(B^*)$. In other words, we fail to pack some of the vertical items since we guessed the widths of the containers only approximately and since our polynomial time approximation algorithm for GAP might not find the optimal packing. We use a similar procedure for the items in I_{hor} where instead of B^* we use B_{hor} in order to place the unassigned items.

▶ **Lemma 3.6.** *In time $n^{O_\varepsilon(1)}$ we can compute a placement for all items in $I_{ver} \cup I_{hor}$ in B^*, B_{hor}, and their corresponding boxes in $\mathcal{B}'$.*

For the medium items we invoke again Lemma 2.11 and we place B_{med} on top of the containers in $\mathcal{B}$ which increases the height of the packing only by 2εOPT.

Finally, we use NFDH again to pack the small items into their corresponding containers in $\mathcal{B}'$, which we denote by $\mathcal{B}'_{small}$, and B_{small}. We need B_{small} due to inaccuracies of NFDH and of our guesses of the container sizes.

▶ **Lemma 3.7.** *In time $n^{O(1)}$ we can compute a placement for all items in I_{small} in $\mathcal{B}'_{small}$ and B_{small}.*

▶ **Theorem 3.8.** *There is a $(3/2 + \varepsilon)$-approximation algorithm for the guillotine strip packing problem with a running time of $n^{O_\varepsilon(1)}$.*

4   Conclusion and Open problems

We were able to show essentially tight approximation algorithms for GSP in both the polynomial and the pseudo-polynomial settings. This was possible due to the structure of the respective optimal packings since they are guillotine separable. However, it is unclear how to obtain such a structured packing in the general case of SP, and the question remains to close the gap between the best approximation guarantee of $(5/3 + \varepsilon)$ and the lower bound of $3/2$. Another interesting open problem related to guillotine cuts is to find out whether there exists a PTAS for the 2D guillotine geometric knapsack (2GGK) problem.

References

1 Anna Adamaszek, Sariel Har-Peled, and Andreas Wiese. Approximation schemes for independent set and sparse subsets of polygons. *J. ACM*, 66(4):29:1–29:40, 2019.

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

3 Nikhil Bansal, Jose R Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: inapproximability results and approximation schemes. *Mathematics of Operations Research*, 31:31–49, 2006.

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

5 Nikhil Bansal, Andrea Lodi, and Maxim Sviridenko. A tale of two dimensional bin packing. In *FOCS*, pages 657–666, 2005.

6 István Borgulya. An eda for the 2d knapsack problem with guillotine constraint. *Central European Journal of Operations Research*, 27(2):329–356, 2019.

7 Adam L. Buchsbaum, Howard J. Karloff, Claire Kenyon, Nick Reingold, and Mikkel Thorup. OPT versus LOAD in dynamic storage allocation. *SIAM J. Comput.*, 33(3):632–646, 2004.

8 Alberto Caprara, Andrea Lodi, and Michele Monaci. Fast approximation schemes for two-stage, two-dimensional bin packing. *Mathematics of Operations Research*, 30(1):150–172, 2005.

9 Henrik I. Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 24:63–79, 2017.

10 Nicos Christofides and Charles Whitlock. An algorithm for two-dimensional cutting problems. *Operations Research*, 25(1):30–44, 1977.

11 François Clautiaux, Ruslan Sadykov, François Vanderbeck, and Quentin Viaud. Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem. *Discrete Optimization*, 29:18–44, 2018.

12   François Clautiaux, Ruslan Sadykov, François Vanderbeck, and Quentin Viaud. Pattern-based diving heuristics for a two-dimensional guillotine cutting-stock problem with leftovers. *EURO Journal on Computational Optimization*, 7(3):265–297, 2019.

13   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*, 9:808–826, 1980.

14   Max A Deppert, Klaus Jansen, Arindam Khan, Malin Rau, and Malte Tutas. Peak demand minimization via sliced strip packing. In *APPROX/RANDOM*, volume 207, pages 21:1–21:24, 2021.

15   Alessandro Di Pieri. Algorithms for two-dimensional guillotine packing problems. Master's thesis, University of Padova, Italy, 2013.

16   Mohammad Dolatabadi, Andrea Lodi, and Michele Monaci. Exact algorithms for the two-dimensional guillotine knapsack. *Computers & Operations Research*, 39(1):48–53, 2012.

17   Fabio Furini, Enrico Malaguti, and Dimitri Thomopulos. Modeling two-dimensional guillotine cutting problems via integer programming. *INFORMS Journal on Computing*, 28(4):736–751, 2016.

18   Waldo Gálvez, Fabrizio Grandoni, Afrouz Jabal Ameli, and Kamyar Khodamoradi. Approximation algorithms for demand strip packing. In *APPROX/RANDOM*, volume 207, pages 20:1–20:24, 2021.

19   Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, Sandy Heydrich, Arindam Khan, and Andreas Wiese. Approximating geometric knapsack via L-packings. *ACM Trans. Algorithms*, 17(4):33:1–33:67, 2021.

20   Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, and Arindam Khan. Improved pseudo-polynomial-time approximation for strip packing. In *FSTTCS*, pages 9:1–9:14, 2016.

21   Waldo Gálvez, Fabrizio Grandoni, Arindam Khan, Diego Ramírez-Romero, and Andreas Wiese. Improved approximation algorithms for 2-dimensional knapsack: Packing into multiple l-shapes, spirals, and more. In *SoCG*, volume 189 of *LIPIcs*, pages 39:1–39:17, 2021.

22   Waldo Gálvez, Arindam Khan, Mathieu Mari, Tobias Mömke, Madhusudhan Reddy Pittu, and Andreas Wiese. A $(2+\epsilon)$-approximation algorithm for maximum independent set of rectangles. *CoRR*, abs/2106.00623, 2021. `arXiv:2106.00623`.

23   P. C. Gilmore and Ralph E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations research*, 13(1):94–120, 1965.

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

25   Sören Henning, Klaus Jansen, Malin Rau, and Lars Schmarje. Complexity and inapproximability results for parallel task scheduling and strip packing. *Theory of Computing Systems*, 64(1):120–140, 2020.

26   Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.

27   Klaus Jansen and Malin Rau. Closing the gap for pseudo-polynomial strip packing. In *ESA*, volume 144 of *LIPIcs*, pages 62:1–62:14, 2019.

28   Klaus Jansen and Guochuan Zhang. On rectangle packing: maximizing benefits. In *SODA*, pages 204–213, 2004.

29   Claire Kenyon and Eric Rémila. A near-optimal solution to a two-dimensional cutting stock problem. *Mathematics of Operations Research*, 25(4):645–656, 2000.

30   Arindam Khan. *Approximation algorithms for multidimensional bin packing*. PhD thesis, Georgia Institute of Technology, 2015.

31   Arindam Khan, Aditya Lonkar, Arnab Maiti, Amatya Sharma, and Andreas Wiese. Tight approximation algorithms for two dimensional guillotine strip packing. *arXiv preprint*, 2022. `arXiv:2202.05989`.

32   Arindam Khan, Arnab Maiti, Amatya Sharma, and Andreas Wiese. On guillotine separable packings for the two-dimensional geometric knapsack problem. In *SoCG*, volume 189, pages 48:1–48:17, 2021.

33   Arindam Khan and Eklavya Sharma. Tight approximation algorithms for geometric bin packing with skewed items. In *APPROX/RANDOM*, volume 207 of *LIPIcs*, pages 22:1–22:23, 2021.

34   Andrea Lodi, Michele Monaci, and Enrico Pietrobuoni. Partial enumeration algorithms for two-dimensional bin packing problem with guillotine constraints. *Discrete Applied Mathematics*, 217:40–47, 2017.

35   Michael L McHale and Roshan P Shah. Cutting the guillotine down to size. *PC AI*, 13:24–26, 1999.

36   Joseph S. B. Mitchell. Approximating maximum independent set for rectangles in the plane. *CoRR*, abs/2101.00326, 2021. `arXiv:2101.00326`.

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

38   János Pach and Gábor Tardos. Cutting glass. In *SoCG*, pages 360–369, 2000.

39   Enrico Pietrobuoni. *Two-dimensional bin packing problem with guillotine restrictions*. PhD thesis, University of Bologna, Italy, 2015.

40   Jakob Puchinger, Günther R Raidl, and Gabriele Koller. Solving a real-world glass cutting problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 165–176. Springer, 2004.

41   Ingo Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *European Symposium on Algorithms*, pages 290–299. Springer, 1994.

42   W Schneider. Trim-loss minimization in a crepe-rubber mill; optimal solution versus heuristic in the 2 (3)-dimensional case. *European Journal of Operational Research*, 34(3):273–281, 1988.

43   Steven S. Seiden and Gerhard J. Woeginger. The two-dimensional cutting stock problem revisited. *Mathematical Programming*, 102(3):519–530, 2005.

44   Daniel Dominic Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Inf. Process. Lett.*, 10(1):37–40, 1980.

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

46   Paul E. Sweeney and Elizabeth Ridenour Paternoster. Cutting and packing problems: a categorized, application-orientated research bibliography. *Journal of the Operational Research Society*, 43(7):691–706, 1992.

47   Lijun Wei and Andrew Lim. A bidirectional building approach for the 2d constrained guillotine knapsack packing problem. *European Journal of Operational Research*, 242(1):63–71, 2015.