Closing the gap for pseudo-polynomial strip packing

Klaus Jansen, Malin Rau
Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Germany
{kj,mra}@informatik.uni-kiel.de

Abstract

We study pseudo-polynomial Strip Packing. Given a set of rectangular axis parallel items and a strip with bounded width and infinite height the objective is to find a packing of the items into the strip which minimizes the packing height. We speak of pseudo-polynomial Strip Packing if we consider algorithms with pseudo-polynomial running time with respect to the width of the strip. It is known that there is no pseudo-polynomial algorithm for Strip Packing with a ratio better than $5/4$ unless $P = NP$. The best algorithm so far has a ratio of $4/3 + \varepsilon$. In this paper, we close this gap: We present an algorithm with approximation ratio $5/4 + \varepsilon$. This algorithm uses a structural result which is the main accomplishment of this paper. This structural result applies to other problem settings as well, which enabled us to present algorithms with approximation ratio $5/4 + \varepsilon$ for Strip Packing with rotations (90 degrees) and Contiguous Moldable Task Scheduling.

1 Introduction

In the Strip Packing Problem, we have to pack a set $I$ of rectangular items into a given strip with width $W \in \mathbb{N}$ and infinite height. Each item $i \in I$ has a width $w(i) \in \mathbb{N}_{\leq W}$ and a height $h(i) \in \mathbb{N}$. A packing of the items is given by a mapping $\rho : I \to \mathbb{N}_{\leq W} \times \mathbb{N}, i \mapsto (x_i, y_i)$ which assigns the lower left corner of an item $i \in I$ to a position $\rho(i) = (x_i, y_i)$ in the strip. An inner point of $i \in I$ (with respect to a packing $\rho$) is a point from the set $\text{inn}(i) := \{(x, y) \in \mathbb{R} \times \mathbb{R} | x_i < x < x_i + w(i), y_i < y < y_i + h(i)\}$. We say two items $i, j \in I$ overlap if they share an inner point (i.e., $\text{inn}(i) \cap \text{inn}(j) \neq \emptyset$). A packing is feasible if no two items overlap and if $x_i + w(i) \leq W$.

In this paper, we study pseudo-polynomial algorithms with respect to the width of the strip. This means we consider algorithms where the width of the strip $W$ is allowed to appear polynomially in the running time. Recently, we were able to show that we cannot find an algorithm with approximation ratio better than $5/4$ unless $P = NP$ [12]. On the other hand, the algorithm with the best ratio so far computes a $4/3 + \varepsilon$ approximation [8, 16]. We manage to prove a strong structural result, which enables us to close the gap to the lower bound, except for an arbitrarily small $\varepsilon$.

Theorem 1. There is a pseudo-polynomial algorithm for the Strip Packing Problem which finds a $(5/4 + \varepsilon)$-approximation in $O(n \log(n)) + n \cdot W^{O_\varepsilon(1)}$ operations.

The structural result applies to other problem settings and, therefore, the algorithmic result can be extended to them. One example is the setting of Strip Packing where we are allowed to rotate the items by 90 degrees. In this setting, the items still have to be placed axis-aligned, but we can decide if the longer or shorter side defines the height of the item.

Theorem 2. There is a pseudo-polynomial algorithm for Strip Packing with rotations which finds a $(5/4 + \varepsilon)$-approximation in $(nW)^{O_\varepsilon(1)}$ operations.
A generalization of the Strip Packing Problem is the Contiguous Moldable Task Scheduling Problem. In this setting, we are given a set of jobs \(J\) and a set of machines \(m\). Each job \(j \in J\) can be scheduled on different numbers of machines given by \(M_j \subseteq \{1, \ldots, m\}\). Depending on the number of machines \(i \in M_j\), each job \(j \in J\) has a specific processing time \(p_j(i) \in \mathbb{N}\). A schedule \(S\) is given by three functions: \(\sigma : J \to \mathbb{N}\), which maps each job \(j \in J\) to a starting time \(\sigma(j)\), \(\rho : J \to \{1, \ldots, m\}\), which maps each job \(j \in J\) to the number of processors \(\rho(j) \in M_j\) it is processed on, and \(\varphi : J \to \{1, \ldots, m\}\), which maps each job \(j \in J\) to the first machine it is processed on. The job \(j \in J\) will use the machines \(\varphi(j)\) to \(\varphi(j) + \rho(j) - 1\) contiguously. A schedule is feasible if each machine processes at most one job at a time and its makespan is defined by \(\max_{j \in J} \sigma(j) + p_j(\rho(j))\). The objective is to find a feasible schedule, which minimizes the makespan.

This problem contains Strip Packing as a special case, when for each job \(j \in J\) it holds that \(|M_j| = 1\). Therefore, we can not hope for a pseudo-polynomial algorithm with a ratio better than \(5/4\) unless \(P = NP\). We managed to adapt the algorithmic result above to find an algorithm with an approximation ratio, which almost matches this bound.

**Theorem 3.** There is a pseudo-polynomial algorithm for the Contiguous Moldable Parallel Tasks Scheduling Problem which finds a \((5/4 + \varepsilon)\)-approximation in \((nm)^{O(1)}\) operations.

We say a job \(j \in J\) is monotone if the work of the job \(w(p_j(i)) := p_j(i) \cdot i\) does not increase if we decrease the number of machines. There is an FPTAS by Jansen and Land [14] for the case that all jobs are monotonic and \(m \geq 8n/\varepsilon\). This algorithm combined with the algorithm from Theorem 3 delivers a polynomial algorithm for the case of monotonic jobs.

**Corollary 1.** There is a polynomial algorithm for Scheduling Monotonic Moldable Parallel Tasks on Contiguous Machines which finds a \((5/4 + \varepsilon)\)-approximation in \(n^{O(1)}\) operations.

**Related Work**

**Strip Packing** Strip Packing is an important NP-hard problem which has been studied since 1980 (Baker et al. [3]). It arises naturally in many settings as scheduling or cutting stock problems in industrial manufacturing (e.g., cutting rectangular pieces out of a sheet of material as cloth or wood). Recently, it also has been applied to practical problems as electricity allocation and peak demand reductions in smart-grids [20, 26, 31].

In a series of papers [2, 3, 5, 6, 9, 11, 17, 21, 27, 28, 29, 30] algorithms with improved approximation ratios have been presented. \(5/3 + \varepsilon\) is the best absolute approximation ratio achieved so far by an algorithm by Harren, Jansen, Prädel, and van Stee [10]. On the other hand by a reduction from the Partition Problem, one can see that it is not possible to find an algorithm with approximation ratio better than \(3/2\) unless \(P = NP\).

However, asymptotic approximation ratios can get better than \(3/2\), and they have been improved in a series of papers [2, 6, 9]. The first asymptotic fully polynomial approximation scheme (in short AFPTAS) was presented by Kenyon and Rémi [21]. It has an additive term \(O(h_{\text{max}}/\varepsilon^2)\), where \(h_{\text{max}}\) is the largest occurring item height. The additive term was improved by Sviridenko [30] and Bougeret et al. [5] to \(O((\log(1/\varepsilon))/\varepsilon)h_{\text{max}}\) simultaneously. Furthermore, Jansen and Solis-Obe [17] presented an asymptotic PTAS with an additive term \(h_{\text{max}}\) at the expense of the running time. Asymptotic algorithms are useful when the maximal occurring item height is small compared to the optimal packing height. However, if the maximal occurring height equals the optimal packing height, these algorithms have an approximation ratio of 2 or even worse. This motivates the search for algorithms with better approximation ratios in expense of the processing time.

Strip Packing can be seen as a Scheduling Problem and is denoted by \(P_{\text{line}j}|C_{\text{max}}\) and is sometimes also called scheduling on non-fragmentable multiprocessor systems [32]. We have given \(W\) machines and
have to schedule parallel jobs on the machines contiguously. Since in realistic instances we can expect the number of machines to be not too large, (e.g., bounded by a polynomial in the number of jobs), it is reasonable to consider pseudo-polynomial algorithms, where we allow $W$ to appear polynomially in the running time. The Partition Problem is solvable in pseudo-polynomial time. Therefore, the lower bound of $3/2$ for absolute approximation ratios does not hold for pseudo-polynomial algorithms. The best approximation ratio has been improved step by step \[ [18, 25, 8, 16]. \] $4/3 + \varepsilon$ is the best absolute approximation ratio achieved so far \[ [8, 16]. \] On the other hand, we can not approximate arbitrary in this scenario. Adamaszek, Kociumaka, Pilipczuk, and Pilipczuk \[ [1] \] proved a lower bound of $12/11$ if $P \neq NP$. This lower bound was improved to $5/4$ by Henning, Jansen, Rau, and Schmarje \[ [12] \] if $P \neq NP$. There are differences in the size of the optimal solutions of the same instances for contiguous task scheduling and the closely related non-contiguous task scheduling $P_{\text{size}j|C_{\text{max}}}$.

These differences were noted by \[ [32] \] and intensively studied by \[ [4]. \] Furthermore notable are the differences in the pseudo polynomial absolute approximation ratio. While for the contiguous case we have a lower bound of $5/4$ if $P \neq NP$, in the noncontiguous case there is a pseudo PTAS \[ [18]. \]

Strip Packing with rotations has been explicitly studied in the following papers \[ [7, 23, 19, 17]. \] Algorithms for Strip packing without rotations using the area of the items to prove their ratio, e.g., NFDH, FFDH \[ [3] \] or Steinberg’s algorithm \[ [29] \], work for Strip Packing with rotations as well. Furthermore, algorithms using 2D Knapsack with area maximization as a subroutine can also be extended to Strip Packing with rotations. On the other hand, the lower bounds of $3/2$ for polynomial and $5/4$ for pseudo polynomial approximation ratios hold for Strip Packing with rotations as well unless $P = NP$.

**Contiguous Moldable Task Scheduling** Moldable jobs are studied for two kinds of jobs, those that need to be scheduled on contiguous machines, and those that do not. Note that due to a reduction from the Partition Problem, algorithms for non-monotonic jobs can not have a ratio better than $3/2$ for both cases, unless $P = NP$. Furthermore for non-monotonic contiguous jobs, we can not find a pseudo polynomial algorithm with ratio better than $5/4$, unless $P = NP$, since this problem contains Strip Packing as a special case. Turek, Wolf and Yu \[ [32] \] presented an algorithm that assigns jobs to numbers of processors and then schedules the fix instance with known algorithms for these scenarios. This algorithms could achieve a $2$-approximation for the non-contiguous case and a $2.5$-approximation for the contiguous case, using Sleator’s algorithm \[ [28] \] as a subroutine. Furthermore, they pointed out that for improved approximations for the fixed processor instances they wold achieve better approximation ratios. More precisely, if the algorithm uses Steinberg’s Algorithm \[ [29] \] as a subroutine instead, it has an approximation ratio of $2$. The running time of these algorithms was improved by Ludwig and Tiwari \[ [22] \] form $O(mn \cdot L(m, n))$ to $O(mn + L(m, n))$, where $O(L(m, n))$ is the running time of the used subroutine for the fixed machine instance. There is a pseudo-polynomial algorithm with ratio $(3/2 + \varepsilon)$ by Mounié et al. \[ [24] \] for monotonic non-contiguous moldable jobs. Jansen and Thöle \[ [18] \] extended the $(3/2 + \varepsilon)$ ratio to non-monotonic contiguous moldable jobs. Furthermore, they presented a pseudo PTAS for non-monotonic non-contiguous moldable jobs. Together with the FPTAS from \[ [14] \] for the case $m \geq 8n/\varepsilon$ this delivers a PTAS for the case of monotonic non-contiguous jobs. Additionally, the running time of the algorithm by Mounié et al. \[ [24] \] is improved to be nearly linear by Jansen and Land \[ [14] \]. A polynomial $(3/2 + \varepsilon)$ approximation algorithm for non-monotonic non-contiguous jobs was presented by Jansen \[ [13] \], which is arbitrary close to the best possible algorithm for this case, unless $P = NP$.

**Methodology and Organization of this Paper**

In the approaches seen before, i.e., in \[ [25, 8 \] and \[ [16], \] a set of critical items is defined. In the $7/5 + \varepsilon$ approximation this set contains all items with height taller than $2/5 \cdot \text{OPT}$, and in the $4/3 + \varepsilon$ approximation, it contains all the items with height larger than $1/3 \cdot \text{OPT}$. In this approach, we define a similar
set, namely all the items with height taller than $1/4 \cdot \text{OPT}$. These sets have in common, that if we place them on top of the optimal packing, we can no longer reach the desired approximation ratio.

All the so far considered approaches show that an optimal solution can be transformed such that the items are arranged in a simple structure, such that all possible structures can be enumerated. For a given structure, one can verify with a dynamic program whether the set of items can be placed into this structure or not. The transformation of the optimal solution consists in all the cases of the following central steps. First, the set of items is partitioned into few types. Then the optimal packing is partitioned into rectangular subareas, such that the subareas contain just items of one (or two) type(s). However, items can overlap the borders of these areas with their longer side but never their shorter side. After this partitioning step, the items in the rectangular subareas are reordered while carefully handling the overlapping items. In this reordering step, some of the items can be discarded from the subareas and are placed on top of the packing afterward. Nevertheless, it will never happen that a critical item is discarded. As a result, the critical items will be contained entirely in the optimal packing area (extended by a factor of at most $(1 + \varepsilon)$) while some of the other items had to be discarded and are placed in the area on top of the optimal packing. Since these items have a height smaller than the critical items, these items do not enlarge the packing too much, such that the desired approximation ratio is not exceeded.

In our approach, we also have these main steps of partitioning the set of items into types and then the packing area into subareas followed by a reordering of the items in the subareas. However, we can no longer guarantee that all the critical items are contained in the optimal packing area. Instead, we introduce a shifting technique, which enables us to place them on three shelves. This shifting technique shifts the items in the optimal packing such that they use the area $W \times (5/4 + \varepsilon)\text{OPT}$. This shifting together with the followed reordering builds the core idea, which enables the improved structure result. In Section 2, we describe this shifting technique and how it leads to a specific structure for the simpler case that just these critical items have to be placed integrally while other items are allowed be partitioned into vertical slices, which do not have to be placed contiguously. In Section 3, we extend this shifting and rearranging strategy to the case that we consider one of the previously described subareas of the packing, where some items can overlap the box border, which have to be handled carefully. Again all critical items in this subarea have to be placed integrally, while the other items in this area can be sliced vertically. In a later step, we can show that we can find a packing of these sliceable items, such that just a few of them have to be sliced. The reordering from Section 3, which also handles overlapping items, enables us to prove the structural result in Lemma 10 from Section 4. However, since by our shifting we already have extended the packing area by $1/4 \cdot \text{OPT}$, we can not simply discard the sliced items and place them on top of the packing. These items have to be placed carefully inside the area $W \times (5/4 + \varepsilon)\text{OPT}$, which complicates matters compared to the approaches before.

Note that optimal solutions of the three considered problems have a similar structure: They consist of rectangular objects placed into a strip. Therefore, modifications to an optimal solution for an instance of the Strip Packing Problem can also be applied to optimal solutions for instances of the other variants, and the structure result holds for these problems as well. Given the structure result from Lemma 10, we can guess the structure and use dynamic programming to place the items or jobs respectively. Thus, we obtain algorithms which find $5/4 + \varepsilon$ approximations for each of the three problems. These algorithms are described in detail in Section 5.

Those who are more interested in the algorithm and less in the proof of the structural result can jump to Section 4 and read to Lemma 10. The proof of Lemma 10 can be skipped, and one can jump to Section 5 to learn about the algorithms. However, the main achievement of this paper is the structure result itself and lies not in the algorithms.
2 A Simple Case

To demonstrate the central new idea which leads to the improved structural result – the shifting and reordering technique – we first consider a relatively simple case. In this simple case, we have to pack items with a tall height integrally, while we are allowed to slice all the other items vertically.

Let a packing with height $H$ be given. We define tall items as the items which have a height larger than $1/4H$ and call the others non-tall. Let us assume that there is an arithmetic grid with $N + 1$ horizontal grid lines with distance $H/N$ such that each tall item starts and ends at the grid lines. For now, we can think of this grid as the integral grid with $H + 1$ grid lines. Later, we can reduce the grid lines by rounding the heights of the items. We are interested in a fractional packing of the non-tall items. Therefore, we replace each non-tall item $i$ by exactly $w(i)$ items with height $h(i)$ and width 1. This step is called slicing.

**Lemma 1.** By adding at most $1/4H$ to the packing height and slicing non-tall items, we can rearrange the items such that we generate at most $3/2N$ containers which contain tall items with the same height only, and at most $9/4N + 1$ container for sliced items.

**Proof.** In this proof, we will present a rearrangement strategy which provides the desired properties. This strategy consists of two shifting steps and one reordering step. In the shifting steps, we shift items in the vertical direction, while in the reordering step we change the item positions horizontally. In the first shifting step, we ensure that tall items intersecting the horizontal lines $1/4H$ or $3/4H$ will touch the bottom or the top of the packing area, respectively. In the second shift, we ensure that tall items not intersecting these lines have a common upper border as well. Last, we reorder the items such that tall items with the same height are positioned next to each other if they have a common upper or lower border.

**First shift.** Note that there is no tall item completely below $1/4H$ or completely above $3/4H$ since each tall item has a height larger than $1/4H$. We shift each tall item $t$ intersecting the horizontal line $1/4H$ down, such that its bottom border touches the bottom of the strip. The sliced items below $t$ are shifted up exactly $h(t)$, such that they are now positioned above $t$. In the same way, we shift each tall item intersecting the horizontal line at $3/4H$ but not the horizontal line at $1/4H$ such that its upper border is positioned at $H$ and shift the sliced items down accordingly.
Introducing pseudo items. At this point, we introduce a set of containers for the sliced items, called pseudo items. We draw vertical lines at each left or right border of a tall item and erase these lines on any tall item, see Figure 1 – ”first shift”. Each area between two consecutive lines which is bounded on top and bottom by a tall item or the packing area and contains sliced items represents a new item called pseudo item. Note that no sliced item is intersecting any box border. When we shift a pseudo item, we shift all sliced items in this container accordingly.

We have one special case. Consider a tall item $t$ with height larger than $3/4 H$. There can be no tall item positioned above or below $t$, and $t$ was shifted down. For these items, we introduce one pseudo item of height $H$ and width $w(t)$ containing $t$ and all sliced items above. Note that each pseudo item has a height, which is a multiple of $H/N$. Furthermore, note that each tall or pseudo item touching the top or the bottom border of the packing area has a height larger than $1/4H$.

Second shift. Next, we do a second shifting step consisting of three steps. First, we shift each tall or pseudo item intersected by the horizontal line at $3/4 H$ but not the horizontal line at $1/4 H$ exactly $1/4 H$ upwards. Second, we shift each pseudo item positioned between the horizontal lines at $1/2 H$ and $3/4 H$, such that their lower border is positioned at the horizontal line $3/4 H$. Last, we shift each tall or pseudo item intersected by the horizontal line at $1/2 H$ but not the horizontal line at $1/4 H$ or $3/4 H$ such that its upper border is positioned at the horizontal line $3/4 H$. After this shifting, no item overlaps an other item since we have shifted the items intersecting the line at $3/4 H$ exactly $1/4 H$, while each item below is shifted at most $1/4 H$.

Fusing pseudo items. In the last step of this second shifting, we will fuse and shift some pseudo items. We want to establish the property that each tall and pseudo item has one border (upper or lower), which touches one of the horizontal lines at $0$, $3/4 H$, or $5/4 H$. At the moment there can be some pseudo items between the horizontal lines $1/4 H$ and $1/2 H$, which do not touch one of the three lines. In the following, we study the three cases where those pseudo items can occur. These items do only occur if there is a tall item touching the bottom of the packing and an other tall item above this item with lower border at or below $H/2$ before the second shifting step. Consider two consecutive vertical lines we had drawn to generate the pseudo items. If a tall item overlaps the vertical strip between these lines, its right and left borders lie either on the strips borders or outside of the strip.

Case 1: In the first case we consider, there are three tall items, $t_1$, $t_2$, and $t_3$ from bottom to top, which overlap the strip. In this scenario $t_1$ must have its lower border at $0$, $t_2$ its upper border at $3/4 H$, and $t_3$ its upper border at $5/4 H$. As a consequence, there are at most two pseudo items: One is positioned between $t_1$ and $t_2$, and the other between $t_2$ and $t_3$. We will stack them, such that the lower border of the stack is positioned at $3/4 H$ and prove that this is possible without overlapping $t_3$. The total height of both pseudo items is $H - h(t_1) - h(t_2) - h(t_3)$. The total area not occupied by tall items is $H - h(t_1) - h(t_2) - h(t_3) + 1/4 H$ since we have added $1/4 H$ to the packing height. The distance between $t_1$ and $t_2$ is at most $1/4 H$ since $t_1$’s lower border is at $0$ and $t_2$’s upper border is at $3/4 H$ and both have a height larger than $1/4 H$. Therefore, the distance between $t_2$ and $t_3$ is at least $H - h(t_1) - h(t_2) - h(t_3)$, see Figure 1 – ”second shift” at the items marked with 1.

Case 2: Now consider the case where there is one tall item $t_1$ touching the bottom, and one tall item $t_2$ with height at least $1/2 H$ touching $5/4 H$. Obviously, $t_2$ has a height of at most $3/4 H$. Furthermore, there is at most one pseudo item, and it has to be positioned between $1/4 H$ and $1/2 H$. We shift this pseudo item up until its bottom touches $1/2 H$, see Figure 1 – ”second shift” at the item marked with 2. This is possible without constructing any overlap, because the distance between $t_1$ and the horizontal line $1/2 H$ is less than $1/4 H$ and therefore the distance between the line $1/2 H$ and the lower border of the tall item is larger than the height of the pseudo item.

After this step, we consider each tall item $t$ with height larger than $1/2 H$ touching $5/4 H$. We generate a new pseudo item with width $w(t)$ and height $3/4 H$, with upper border at $5/4 H$ and lower border at $1/2 H$, containing all pseudo items below $t$ touching $1/2 H$ with their lower border.
Case 3: In the last case we consider, there are two tall items $t_1$ and $t_2$ and two pseudo items; one of the items $t_1$ and $t_2$ touches the top of the packing or the bottom, while the other ends at $3/4H$. Hence, the distance between the tall items has to be smaller than $1/4H$. Furthermore, one of the pseudo items has to touch the top or the bottom of the packing while the other is positioned between $t_1$ and $t_2$. Since the distance between $t_1$ and $t_2$ is less than $1/4H$ one of the distances between the packing border and the lower border of $t_1$ or the upper border of $t_2$ is at least $H - h(t_1) - h(t_2)$. Therefore, we can fuse both pseudo items by shifting the one between $t_1$ and $t_2$ such that it is positioned above or below the other one, see Figure 1 – ”second shift” at the items marked with 3.

Consequence: After the shifting and fusing, each tall or pseudo item touches on of the horizontal lines at 0, $3/4H$ or $5/4H$.

Reordering the items. In the last part of the rearrangement, we reorder the items horizontally to place pseudo and tall items with the same height next to each other. In this reordering step, we move certain items to certain areas. To do this, we take vertical slices of the packing and move them in the strip. A vertical slice is an area of the packing with width one and height of the considered packing area, i.e. $5/4H$ in this case. If we rearrange some of these slices, it will never happen that two items overlap. However, it can happen, that some tall items are placed fractionally afterwards. This will be fixed in later steps.

Area 1: First, we will extract all vertical slices containing (pseudo) items with height $H$. Then, shifting all the remaining vertical slices to the left as much as possible, we create one box for pseudo items of height $H$ at the right, see Figure 1 – ”reordered packing” Area 1. In this area, we sort the pseudo items such that the pseudo items containing tall items with the same height are placed next to each other. In this step we did not place any tall item fractionally.

Area 2: Afterward, we take each vertical slice containing a (pseudo) item with height at least $1/2H$ touching the horizontal line at $5/4H$. Remember, there might be pseudo items containing a tall item $t$ with height between $1/2H$ and $3/4H$. We shift these slices to the left of the packing and sort them in descending order of the tall items height $h(t)$, see Figure 1 – ”reordered packing” Area 2. Afterward, we sort the pseudo items below these tall items, which are touching $1/2H$ with their bottom in ascending order of their heights, which is possible without generating any overlapping. In this step, it can happen that we slice tall items which touch the bottom of the strip. We will fix this in one of the following steps, when we consider Area 5.

Area 3: Next, we look at vertical slices containing (pseudo) items $t$ with height at least $1/2H$ touching the bottom of the strip. We shift them to the right until they touch the Area 1 and sort these slices in ascending order of the heights $h(t)$, see Figure 1 – ”reordered packing” Area 3. Note that there are no pseudo or tall items with upper border at $3/4H$ in these slices. In this step, it can happen that we slice tall items touching the top of the packing. This will be fixed in the next step.

Area 4: Look at the area above $3/4H$ and left of the Area 2 but right of Area 1, see Figure 1 – ”reordered packing” Area 4. In this area no item overlaps the horizontal line $3/4H$. Therefore, we have a rectangular area where each item either touches its bottom or its top and no item is intersected by the area’s borders. In [25] it was shown that, in this case, we can sort the items touching the line $3/4H$ in ascending order of their height and the items touching $3/4H$ in descending order of heights and no item will overlap an other item. Since now all items with the same height are placed next to each other, we have fixed the slicing of tall items on the top of the strip.

Area 5: In the last step, we will reorder the remaining items. Namely the items touching the bottom of the strip left of Area 3 and the items touching the horizontal line at $3/4H$ with their top between Area 2 and Area 3. The items touching the bottom are sorted in descending order of their height and the items touching the horizontal line at $3/4H$ are sorted in ascending order of their heights.

Claim. After this step no item overlaps another.
First, note that the items touching \(5/4H\) having height larger than \(1/2H\) have a height of at most \(3/4H\). Therefore, no item touching the bottom having height at most \(1/2H\) can overlap with these items. Furthermore, note that before the reordering no item was overlapping an other. Let us assume there are two items \(b\) and \(t\), which overlap at a point \((x,y)\) after this reordering. Then all items left of \(x\) touching \(3/4H\) have their lower border below \(y\), while all items touching the bottom left of \(x\) have their upper border above \(y\). Therefore, at every point left and right of \((x,y)\) in the Area 5 there is an item overlapping it. Hence, the total width of items overlapping the horizontal line \(y\) is larger than the width of the Area 5. Since we did not add any items, in the original ordering there would have been items overlapping each other already, a contradiction. As a consequence in this new ordering, no two items overlap, which concludes the proof of the claim.

Analyzing the number of constructed boxes. In the last part of this proof, we analyze how many boxes we have created. Each tall item with height at least \(3/4H\) touches the bottom and we create at most one box in Area 1 for each height. Therefore, we create at most \(N/4\) boxes for these items. Each tall item of height between \(1/2H\) and \(3/4H\) either touches the bottom or the horizontal line \(5/4H\). On each of these lines, we create at most one box for items with the same height. Therefore, we create at most \(2N/4\) boxes for these items. Last, each tall item with height larger than \(1/4H\) but smaller than \(1/2H\) either touches the bottom of the packing, the horizontal line \(3/4H\) or the horizontal line \(5/4H\). At each of these lines, we create at most one box for each height. Therefore, we create at most \(3N/4\) of these boxes. In total, we create at most \(3N/4\) boxes for tall items.

Let us consider the number of boxes for sliced items. Each pseudo item’s height is a multiple of \(H/N\). Therefore we have at most \(N\) different sizes for pseudo items. There are at most 4 boxes for each height less than \(1/4H\). One is touching \(H\) with its top border in Area 1, one is touching \(3/4H\) with its bottom border in Area 4, one is touching \(3/4H\) with its top border in Area 5, and one is touching \(1/2H\) with its bottom border in Area 2. Furthermore, there are at most 3 boxes for each size between \(1/4H\) and \(1/2H\). One is touching \(5/4H\) with its top border in Area 4, one is touching \(3/4H\) with its top border in Area 5, and one is touching 0 with its bottom border in Area 5. Additionally, there are at most 2 boxes for each pseudo item size larger than \(1/2H\). One is touching \(5/4H\) with its top border in Area 2, the other is touching 0 with its bottom border in Area 3. Last there is only one pseudo item with height larger than \(3/4H\) in Area 1. It has height \(H\). If the grid is arithmetically we have at most \(N/4\) sizes with height at most \(1/4H\), \(N/4\) sizes between \(1/4H\) and \(1/2H\) and at most \(3N/4\) sizes between \(1/2H\) and \(3/4H\). Therefore, we create at most \(4 \cdot 1/4N + 3 \cdot 1/4N + 2 \cdot 1/4N + 1 = \frac{9}{4}N + 1\) boxes for sliced items.

\[\square\]

3 Reordering in the General Case

In the structural result, all items have to be placed integrally; thus, we can not slice all non-tall items. Nevertheless, we may still slice certain narrow items, because for them we have techniques to place them integrally afterward. We call these sliceable items vertical items. As proven in [16], it is possible to partition the packing area into a constant number of rectangular subareas, called boxes, such that boxes containing tall items will contain tall and vertical items only. In this section, we will consider such boxes and show that it is possible to reorder the items in these boxes similarly as in Section 2. A difficulty here is that up to three tall items can overlap the left and right box border. Since we do not want to slice these items, we fix their position and call them unmovable items. These unmovable items complicate the reordering in the box. We overcome this difficulty, by a more careful reordering of the items, while the shifting steps remain the same.

The most interesting boxes are the boxes which have a height of at least \(3/4H\). In these boxes, there can be up to three tall items above each other, while in smaller boxes there can be at most two. How to handle boxes with at most two tall items on top of each other was already proven in [16]. They are
partitioned into fewer sub boxes. While the boxes larger than $\frac{1}{2}H$ still need an extra height of $\frac{1}{4}H$, for the boxes smaller than $\frac{1}{2}H$ their original height is sufficient.

The reordering in this section will be used in the proof of the structure Lemma 10. When we use the reordering the boxes already have certain properties. Therefore in the following, we can assume that boxes with height larger than $\frac{3}{4}H$ can be extended by $\frac{1}{4}H$. Furthermore, let $S(B)$ be the y-coordinate of the lower box border. We assume that no tall item overlaps the left or right box border at or above $S(B) + H' - \frac{1}{4}H$. The details why we can assume these properties can be found in the proof of Lemma 10.

In the reordering strategy discussed in this section, we will use a version of a result in [16] on how to reorder boxes in which each item either touches the top or the bottom of the box, and there are up to two unmovable items at each box border:

**Lemma 2 ([16]).** Let $B$ be an area where each item either touches the bottom or the top of this area, with at most two unmovable items on each side. Let $S_P$ be the number of heights of pseudo items in this area, $S_T$ be the number of heights of tall items in this area and $S_{P\cup T}$ be the number of heights in $P \cup T$. It is possible to rearrange the items in this area creating at most $4S_T S_{P\cup T}$ boxes for tall items plus one for each unmovable item and at most $4S_P S_{P\cup T}$ boxes for pseudo items plus the pseudo items for extending the unmovable items.

In this section, we still assume that all tall items are placed on an arithmetic grid with $N+1$ horizontal grid lines with distance $H/N$. Furthermore, we assume that the box also starts and ends at these grid lines. Let $S(B)$ be the y-coordinate of the lower box border.

**Lemma 3.** Let $B$ be a box with height $H' > \frac{3}{4}H$, such that no tall item overlaps the left or right box border at (or above) $S(B) + H' - \frac{1}{4}H$. By adding at most $\frac{1}{4}H$ to $B$’s height, we can rearrange the items in $B$ such that we generate at most $2N^2 + 15N/4 + 8$ boxes for tall and at most $4N^2 + 31N/4 + 5$ boxes for vertical items without moving the unmovable items. The vertical items are sliced while each tall item is placed as a whole.

**Proof.** In this proof, we present a new reordering strategy for the items in these boxes. Let $H'$ be the height of $B$. For convenience, we will assume that the lower border of $B$ is at 0. If not, we shift all horizontal lines accordingly. Notice that there are at most two tall items overlapping the left or right box border, since we assumed that there is no tall item overlapping the border at $H' - \frac{1}{4}H$. In the first step, we shift all movable items according to the first and second shifting step seen in the proof of Lemma 1. However, the reordering works differently than before. If there are items taller than $\frac{1}{2}H$ touching the top of the box, we find the leftmost item $l$ and the rightmost item $r$ of them. We introduce three areas: one left of $l$, one between $l$ and $r$ and one right of $r$. While we reorder the leftmost and the rightmost area with known techniques, we need a new trick to reorder the middle part.

**Shifting the items.** We shift analogously as in the simple case, see Figure 2. However, we have to be careful with the overlapping items. First, we shift each movable item crossed by the line $\frac{1}{4}H$ down to the bottom of the box. Afterward, we shift each movable item crossed by the line $H' - \frac{1}{4}H$ to the top of the box. We introduce pseudo items as described in the proof of Lemma 1, with the difference that each tall item $t$ with height larger than $3H'/4$ generates a pseudo item with height $H'$ and width $w(t)$.

Let us now consider the unmovable items on the left box side. There can be two of these, one overlapping the box at $\frac{1}{4}H$, the other at $\frac{1}{2}H'$. If there is just one item, we extend it to the bottom of the strip, generating one unmovable pseudo item. If there are two items, $t_1$ at $\frac{1}{4}H$ and $t_2$ at $\frac{1}{2}H'$, we extend $t_1$ to the bottom. Then, depending on which of the items $t_1$ or $t_2$ has its right border father on the left, we extend $t_1$ to the bottom of $t_2$ or $t_2$ to the top of $t_1$, see Figure 2 at the hatched areas. We do the same on the right side of the box.
Next, we do the second shifting step, see Figure 2. Each tall and pseudo item cut by the line $H' - \frac{1}{4}H$ is shifted up exactly $\frac{1}{4}H$. Remember, there is no unmovable item intersecting this line. We shift each pseudo item between the lines $H' - \frac{1}{4}H$ and $H' + \frac{1}{4}H$ such that its bottom touches $H' - \frac{1}{4}H$. Afterward, we shift each unshifted movable tall and pseudo item crossed by the line $\frac{1}{2}H'$ such that its top touches $H' + \frac{1}{4}H$. Again, no item overlaps another after this shift.

Last, we will fuse the pseudo items as described in Lemma 1. The fusion is possible, since the considered distance in each of the Cases 1 to 3 is at most $\frac{1}{4}H$ too. After this fusion, we can assume that each item $t$ with height larger than $\frac{1}{2}H'$ touching $H' + \frac{1}{4}H$ has a height of exactly $\frac{1}{2}H' + \frac{1}{4}H$, see Case 2 in the proof of Lemma 1.

Furthermore, we can assume that each item $t$ touching the bottom with height taller than $\frac{1}{2}H'$ has height $H' - \frac{1}{4}H$: There can be at most two items above $t$, one tall item and one pseudo item. The pseudo item has its lower border at $H' - \frac{1}{4}H$. Therefore, we can extend the item $t$ to the horizontal line $H' - \frac{1}{4}H$.

After this shift, each movable item has one border at one of the following horizontal lines $0$, $H' - \frac{1}{4}H$, or $H' + \frac{1}{4}H$. Furthermore, only (pseudo) items with height $\frac{1}{2}H' + \frac{1}{4}H$ or larger are crossing the line $H' - \frac{1}{4}H$.

**Reordering.** Let us for simplicity assume that there is no (pseudo) item with height $H'$ in $B$. Later we will see, what happens if there are any of these items. In the following, we will reorder the items step by step, by considering a constant number of smaller subareas of the box. For this purpose, let $l$ be the leftmost and $r$ the rightmost pseudo item with height $\frac{1}{2}H' + \frac{1}{4}H$ touching $H' + \frac{1}{4}H$. In the following steps, we will do the same things we do next to $l$ in the same way but mirrored next to $r$.

**Area $B_{l,1}$:** Left of $l$ in the box, there is no item intersecting the horizontal line $H' - \frac{1}{4}H$. Therefore, each item left of $l$ above $H' - \frac{1}{4}H$ either touches $H' - \frac{1}{4}H$ with its lower border or $H' + \frac{1}{4}H$ with its upper border. Since there is no item intersecting the left box border, we can sort the items left of $l$ touching $H' + \frac{1}{4}H$ in descending order and the items touching $H' - \frac{1}{4}H$ in ascending order of their heights, without constructing any overlap. The same holds for the right side of $r$. We call these areas $B_{l,1}$ and $B_{r,1}$, see Figure 3.

**Area $B_{l,2}$:** We draw a vertical line at the left border of $l$ to the bottom of the box. If this line cuts...
a tall item $l_b$ at the bottom, it defines a new unmovable item. Let us consider the area between the line and the left box border below $H' - \frac{1}{4}H$. We call this area $B_{l,1}$. In $B_{l,1}$ each item either touches 0 or $H' - \frac{1}{4}H$ and on each side there are at most two tall unmovable items. We reorder these boxes with the techniques from Lemma 2. We do the same on the right of $r$, see Figure 3.

Cases for $l$ and $r$: If $l$ and $r$ do not exist there are no items overlapping the horizontal line $H' - H/4$ and we can partition the box in two areas $B_1$ and $B_2$. We reorder $B_1$ as described for $B_{l,1}$ and $B_2$ as described for $B_{l,2}$. In the case that $l$ equals $r$ we introduce $B_{l,1}$, $B_{l,2}$, $B_{r,1}$ and $B_{r,2}$ as described, and order the tall items completely below $l$ such that items with the same height are positioned next to each other.

Area $B_{l,3}$: If the right border of $l_b$ is right of the right border of $l$, we draw a vertical line at the right border of $l_b$, called $L_1$. If $L_1$ intersects a tall item with upper border at $H' - \frac{1}{4}H$, we call this item $l_m$. Left of $L_1$ and right of $l$, we shift up each item touching $H' - \frac{1}{2}H$ with its top (the item $l_m$ inclusively), such that its lower border touches $\frac{1}{2}H'$, and shift down each pseudo item touching $H' - \frac{1}{4}H$ with its lower border, such that it touches $\frac{1}{2}H'$ with its upper border. All pseudo items right of $L_1$ above $l_m$ are shifted, such that they touch the top of $l_m$ with their bottom, see Figure 3. Note, that no pseudo item is intersected by the line $L_1$.

After this shift no item overlaps an other, since the vertical gap between the tall items where we do not place the pseudo item is at most $\frac{1}{4}H$ and therefore the residual space is large enough to place it. Let $I_{l,1/2H'}$ be the set of shifted (pseudo) items touching now $\frac{1}{2}H'$ with their bottom. All the items in $I_{l,1/2H'}$ have a height of at most $\frac{1}{2}H'$. The area left of $L_1$ and right of the left border of $l$ below $\frac{1}{2}H'$ is called $B_{l,3}$. This area contains pseudo items touching $\frac{1}{2}H'$ and a part of $l_b$ at the bottom. We sort the pseudo items above $l_b$ touching $\frac{1}{2}H'$ in descending order of their heights.

On the other hand, if the right border of $l_b$ is left of the right border of $l$, we introduce the line $L_1$ too, but do not shift any item. On the right of $r$, we introduce the same lines and areas named $R_1$ and $B_{r,3}$.
respectively. It can happen that \( l_b \) equals \( r_b \), or one of the lines \( L_1 \) or \( R_1 \) intersects with \( r \) or \( l \) respectively, or that \( L_1 \) equals \( R_1 \). In each of these cases, there is no item with height larger than \( H'/2 \) touching the bottom of the box between the lines \( L_1 \) and \( R_1 \). If there is no such item, we shift all the items touching \( H' - 1/4H \) with their top between \( L_1 \) and \( R_1 \), such that they touch \( 1/2H' \) with their bottoms and the pseudo items touching \( H' - 1/4H \) with their bottom such that they touch \( 1/2H' \) with their top. Now there is no item intersecting the horizontal line \( 1/2H' \) (since there is no item with height larger than \( H' \) touching the horizontal line 0 between \( L_1 \) and \( R_1 \)). Hence, we can sort the items above \( 1/2H' \) between \( l \) and \( r \) by their heights as well as the items below \( 1/2H' \). After this step, we do not need any further reordering.

The item \( i \): We now consider the case that none of the previous named cases has occurred and that we need further reordering. We want to reorder the items of height \( 1/2H' + 1/4H \) touching \( H' + 1/4H \), such that they build two blocks, one next to \( l \) and one next to \( r \). To make this reordering possible, we have to define a border between \( l \) and \( r \) such that all these items left of this border are shifted to the item \( l \) while all these items right of this border are shifted to the item \( r \). Let \( i \) be an item of height \( H' - 1/4H \) touching the bottom between \( L_1 \) and \( R_1 \). This item defines the border between \( l \) and \( r \). If \( i \) does not exist, there is no item overlapping the horizontal line \( 1/2H' \) and touching the bottom. What we do in this case is covered in the previous paragraph.

Area \( B_{l4} \) and Area \( B_{l5} \): Consider (pseudo) items with height \( 1/2H' + 1/4H \) touching \( H' + 1/4H \), right of \( l \) and left of \( i \). Note, that none of these items is positioned above \( i \). We shift those items left of \( i \) to the left until they touch \( l \) and those items right of \( i \) to the right until they touch \( r \). All other items with parts above \( 1/2H' \) are shifted to the right or left accordingly, see Figure 4. We sort the items with height \( 1/2H' + 1/4H \) such that the pseudo items containing tall items with an equal height are positioned next to each other. The area containing these items left of \( i \), \( l \) inclusively, is called \( B_{l4} \).

While we shift the items with height \( 1/2H' + 1/4H \) touching \( H' + 1/4H \) such that they are close to \( l \) and \( r \), we shift all the items between \( l \) and \( r \) with height \( H' - H/4 \) touching the horizontal line 0, such that they are next to \( i \) and shift the other items to the left or right accordingly. These items form a new area around \( i \) called \( B_5 \).
In this step it can happen that items touching $H' - \frac{1}{4}H$ with their top are intersecting items touching 0 with their bottom. We will fix this in a later step.

**Area $B_{l,6}$:** Note that the items in the set $I_{1,\frac{1}{2}H'}$ are now placed next to each other (before it was possible that items with height $\frac{1}{2}H' + \frac{1}{4}H$ where positioned between them). In addition, there is no item touching $H' + \frac{1}{4}H$ above an item touching $H' - \frac{1}{4}H$ with their bottom, which was not above this item before. Furthermore, the total width of items with bottom border above $\frac{1}{4}H$ and below $\frac{1}{2}H'$ between $L_1$ and the right of $i$ has not changed.

If $l_m$ exists, we draw a vertical line $L_2$ at the right of $l_m$ and a vertical line $L_3$ at the left of $l_m$. Let $l_{t,r}$ and $l_{t,l}$ be the tall items touching $H' + \frac{1}{4}H$ intersected by this line if there are any. We look at the area left of $L_3$ and right of $B_{l,4}$, which is bounded at the top by $H' + \frac{1}{4}H$ and at the bottom by $\frac{1}{2}H'$. We call this area $B_{l,6}$. In this area, each item touches the bottom or the top, and there is at most one item $l_{t,l}$ intersecting the border, see Figure 4. We use the reordering in Lemma 2 to reorder the items in $B_{l,6}$.

**Area $B_{l,7}$:** The area above $l_m$ is called $B_{l,7}$, see Figure 4. In this area, all items are touching $H' + \frac{1}{4}H$ or the top of $l_m$. All the items touching $l_m$ with their bottom (and not $H' + \frac{1}{4}H$ with their top) are pseudo items. We order the items touching $H' + \frac{1}{4}H$ in ascending order of their heights and move the pseudo items below with them. Now, we look at the overlapping items $l_{t,r}$ and $l_{t,l}$. We move items with the height $h(l_{t,r})$ and $h(l_{t,l})$ next to these overlapping items. This generates three areas for pseudo items. One is positioned below the first overlapping item together with the items with the same height the, second below the other overlapping item together with the items with the same height, and the last between these areas. In each of these areas, we sort the pseudo items in descending order of their height.

The areas $B_{l,6}$ and $B_{l,7}$ exist only if $l_m$ exists. If $l_m$ does not exist, we introduce the vertical line $L_2$ at the left border of the area $B_{l,4}$. We introduce $R_2$ and the areas $B_{r,6}$ and $B_{r,7}$ analogously on the left of $r$.

**Area $B_8$:** Look at the area above $H' - \frac{1}{4}H$ right of $L_2$ and left of $R_2$, see Figure 5. We call this area $B_8$. There are at most two unmovable items overlapping this area. One item $l_{t,r}$ on the left touching
$H' + \frac{1}{4}H$ and one item $r_{t,1}$ on the right touching $H' + \frac{1}{4}H$. Since $B_8$ does not contain any item of height $\frac{1}{2}H' + \frac{1}{4}H$ or items from the sets $I_{1,4}^{1/2H'}$ or $I_{r,1/3H'}$, each item touches either the top or the bottom of this area. Furthermore, all items touching the bottom are pseudo items. Therefore, we can sort the items in this area as they are sorted in area $B_{l,7}$.

**Area $B_{l,9}$:** Last, we have to look at the items on the bottom between $L_1$ and $R_1$ as well as at the items touching $H' - \frac{1}{4}H$ with their top between $L_2$ and $R_2$, see Figure 5. We consider the items touching the bottom between $L_1$ and the left border of $B_5$ and the items touching $H' - \frac{1}{4}H$ with their top between $L_2$ and the left border of $B_5$. The area containing these items is called $B_{l,9}$. In $B_{l,9}$, we sort all items touching $H' - \frac{1}{4}H$ in ascending order of their heights and the items at the bottom in descending order of their heights, such that the tallest on the bottom touches $l_b$ and the smallest touches the area $B_5$. We do the same but mirrored on the right side of $i$ in the area $B_{r,9}$.

**Claim.** After this step there is no item which overlaps another in the area $B_{l,9}$.

First, no item from the bottom will overlap the items with height $\frac{1}{2}H' + \frac{1}{4}H$ from the top since their lower border is at $\frac{1}{2}H'$ and the items below have a height less than $\frac{1}{2}H'$ (otherwise they would be contained in area $B_5$).

Let us assume there is an item $b$ from the bottom intersecting an item $t$ from the top at an inner point $(x,y)$ in the area $B_{l,9}$. As a consequence, for each $x'$ larger than $x$ up to the left border of $B_5$, the point $(x',y)$ is overlapped by an item touching $H' - \frac{1}{4}H$, and for each $x' \leq x$ but left of $L_1$ the point $(x',y)$ is overlapped by an item from the bottom of the box. Note that the total width of items with lower border below $y$ and above $\frac{1}{4}H$ between $L_1$ and the left border of $i$ has not changed after the shifting of items with height $\frac{1}{2}H' + \frac{1}{4}H$ on the top of the box (the items left of $l_m$ have their lower border at $H'/2$). Additionally, the total width of items touching the bottom of the box with upper border above $y$ in this area has not changed either. Therefore, the total width of items overlapping the horizontal line at $y$ in this area is larger than the width of this area. As a result the items must had have an overlapping before the first horizontal shift – a contradiction. Hence, there is no item overlapping another item in this area, which concludes the proof of the claim.

**Items with height $H'$:** Last, let us consider the case that we have (pseudo) items with height $H'$. In this case we choose one of the items with height $H'$ and move all the other items with this height, such that they are positioned next to this item, without slicing any tall item. Then these items form an area $B_{10}$, which just contains items of height $H'$. We sort the pseudo items containing tall items such that tall items with the same height are placed next to each other. Then we will search for $l$ left of this area and for $r$ right of this area. This area divides the box in two parts and represents the splitting item $i$. In this case the box $B_5$ is split into two parts $B_{l,8}$ and $B_{r,8}$, as well as the area $B_5$ is divided into $B_{l,5}$ and $B_{r,5}$. All the following steps are done as described above.

**Analyzing the number of constructed boxes.** In the worst case we have (pseudo) items with height $H'$ and both $l$ and $r$ exist. Furthermore, the left border of $l_b$ should be right of the left border of $l$ as well as the left border of $r_b$ should be left of $r$. Until now we did not need the assumption, that the tall items are placed on an arithmetic grid. However, to count the generated boxes, it is convenient to make this assumption. First, we will analyze the number of boxes for tall items we generate.

**Claim.** The number of boxes for tall items is bounded by $2N^2 + 15N/4 + 8$.

In the areas $B_{l,1}$ and $B_{r,1}$ there are tall items with heights between $\frac{1}{4}H$ and $\frac{1}{2}H$ on the top of the box. For each of these sizes we generate at most one box in each area. Therefore, both contain at most $N/4$ boxes for tall items.

In the boxes $B_{l,2}$ and $B_{r,2}$, we create at most one box for each item height larger than $\frac{1}{2}H'$ and lower than $H'/3$. There are at most $N/4$ sizes. For the other occurring sizes we create by Lemma 2 at most $4S_T S_{T\cup P}$ boxes total, since there are at most three unmovable items overlapping this area.
We have $S_T \leq N/4$ since the tall items have heights between $1/4H$ and $1/2H'$, $S_P \leq N/2$ since they have heights smaller than $1/2H'$, and $S_{T \cup P} \leq N/2$ as a consequence. Therefore, we create at most $4N^2 + 3 = N^2/2 + N/4 + 3$ boxes for tall items in each of the areas $B_{l,2}$ and $B_{r,2}$.

The area $B_{l,3}$ just contains the item $l_b$ as a tall item. Since this item overlaps the area $B_{l,2}$, we have already counted this item. The area $B_{l,4}$ contains just tall items with height between $1/2H'$ and $H'-1/4H$. For each size we create one box. Therefore, we create at most $N/4$ boxes for tall items in this area. The area $B_{l,5}$ contains the tall items with height between $1/2H'$ and $H' - 1/4H$. For each of these sizes we create at most one box, resulting in at most $N/4$ boxes in this area. In the areas $B_{l,6}$ and $B_{r,6}$ each tall and pseudo item has a size of less than $1/2H'$. Analogously to the boxes $B_{l,2}$ and $B_{r,2}$ we create by Lemma 2 at most $N^2/2 + 1$ boxes for tall items per area $B_{l,3}$ and $B_{r,3}$, since there is at most one overlapping item. The area $B_{l,7}$ or $B_{r,7}$ is the area containing $l_m$ or $r_m$ respectively. Above $l_m$ and $r_m$ we create at most $N/4$ boxes for tall items each, since the tall items have a height of at most $1/2H'$ and at least $1/4H$. The box for the item overlapping $L_3$ is already counted.

$B_8$ is divided into two boxes, if we have (pseudo) items with height $H'$. In each of these parts each tall item has height at most $1/2H'$ and we create one box per item size. Therefore we create at most $N/4$ boxes in this area in each part. The boxes for the items overlapping $L_2$ or $R_2$ are already counted for area $B_{l,7}$. We consider now the areas $B_{l,9}$ and $B_{r,9}$. In these areas, all items have height of at most $1/2H'$ and for each item height we create one box at the bottom and one box at $H'-1/4H$. Therefore, we create at most $2N/4$ boxes in each area. Last, we create at most one box for each item with height larger than $H'-1/4H$ resulting in at most $N/4$ boxes for these items.

In total the number of generated boxes is bounded by $2(N/4 + N^2/2 + N/4 + 3 + N/4 + N^2/2 + 1 + N/4 + 2N/4 + N/4) + N/4 = 2N^2 + 15N/4 + 8$, which concludes the proof of the claim.

Let us now consider the number of boxes for vertical items.

Claim. The number of boxes for vertical items is bounded by $4N^2 + 31N/4 + 5$.

In the areas $B_{l,1}$ and $B_{r,1}$ we have at most $N/4$ boxes for items touching the bottom, since they have height of at most $1/4H$ and at most $N/4$ boxes for items touching the top, since they have height of at least $1/4H$ and at most $1/2H'$. Therefore, in each of the areas $B_{l,1}$ and $B_{r,1}$ we generate at most $N/2$ boxes. In the areas $B_{l,1}$ and $B_{r,1}$ the pseudo items touching the bottom have sizes between $1/4H$ and $1/2H'$ and the items touching the top have sizes up to $1/2H'$. By Lemma 2 we generate at most $4S_T S_P S_{T \cup T} \leq 4N^2 N = N^2$ boxes plus the two boxes for extending the unmovable items in each area. Therefore, in these both areas $B_{l,2}$ and $B_{r,2}$, we create at most $N^2 + 2$ boxes for pseudo items.

In the area $B_{l,3}$ and $B_{r,3}$ above the items $l_b$ and $r_b$ respectively, there are pseudo items with heights up to $1/4H$. For each size we generate at most one box. Therefore, we generate at most $N/4$ boxes in each of these areas. In the area $B_{l,4}$ below, the items with height larger than $1/2H'$ we have areas for pseudo items with height at most $1/4H$. We have two blocks of these items, one at $l$ the other at $r$. In each of these areas we create at most $N/4$ boxes for these items. Furthermore, there can be pseudo items with heights between $1/2H'$ and $H'-1/4H$ for each of these heights we create at most one box resulting in $N/4$ boxes for these items in each area $B_{l,4}$ and $B_{r,4}$. In the areas $B_{l,6}$ and $B_{r,6}$ the pseudo items have heights between $1/4H$ and $1/2H'$ on the top and heights up to $1/2H'$ on the bottom. Therefore, by Lemma 2 we generate at most $N^2$ boxes analogously to the boxes $B_{l,2}$. Here, we do not create an other pseudo item, since the item $l_{i,1}$ already touches the top of the area. Therefore, we generate at most $N^2$ boxes in each of the areas $B_{l,5}$ and $B_{r,5}$. In the area $B_{l,7}$ above $l_m$, we have at most three areas for pseudo items touching $l_m$ with their lower border. these items have a height of at most $1/4H$. therefore we create at most $3N/4$ for these pseudo items. Furthermore, the pseudo items touching $H'+1/4H$ with their top have a height between $H'/4$ and $1/2H'$, for each height we generate at most one box. Therefore, we create at most $N/4$ boxes for these items. In total we generate at most $N$ boxes for pseudo items in the areas $B_{l,7}$ and $B_{r,7}$ each.
In the area $B_8$ pseudo items with height up to $1/4H$ touch the bottom and items with sizes between $1/4H$ and $1/2H'$ are touching the top. For each size we generate at most one boxes. Therefore, in $B_8$ we generate at most $N/2$ boxes. $B_8$ can be split in two by the items with height $H'$. Therefore, we have to count the boxes in $B_8$ twice. In the area $B_{l,9}$ the tall items on the bottom have height between $1/4H$ and $3H'/4$. For each of these sizes we create at most one box, summing up to at most $N/4$. On the top of this area the pseudo items have heights up to $1/2H'$ and we create one box per size, creating at most $N/2$ boxes. Therefore in the areas $B_{l,4}$, we have at most $3N/4$ boxes total. In the area $B_{l,5}$ above the tall items with height between $1/2H'$ and $H' − 1/4H$ there are no pseudo items, they were shifted up, to have their lower border at $H' − H/4$. This area contains just pseudo items with height between $1/2H'$ and $H' − 1/4H$ and for each size we create at most one box, hence at most $N/4$ boxes. Last, we consider the items with height larger than $3H'/4$ in area $B_{l,10}$. Above these items there can be pseudo items with heights up to $1/4H$. For each size we create at most one box. Therefore we create at most $N/4$ boxes above these items. Furthermore there can be at most one box contain a pseudo item with height $H'$. In total we create at most $2(N/2 + N^2/4 + N/4 + N/2 + N^2 + N + N/2 + 3N/4 + N/4) + N/4 + 1 = 4N^2 + 31N/4 + 5$ boxes for vertical items, which concludes the proof of the claim and therefore the proof of the lemma.

In the case that the considered box $B$ has a height of at most $3/4H$, there are at most two tall items on top of each other. In this box we can shift the tall items to the top and to the bottom and generate pseudo items as described in the previous proof. Pseudo items, which are positioned vertically between two tall items are removed and placed in a later step. Then all tall and pseudo items have height $h(B)$ or a height between $1/4H$ and $h(B) − 1/4H$. Therefore, tall and pseudo items have at most $N/4$ different heights in this area and the smallest items touching the top or the bottom have a height of at least $1/4H$. Furthermore, the difference between heights is at least $H/N$. Therefore, we can conclude the following Lemma from what is proven in [16].

**Lemma 4 ([16]).** Let $B$ be a box with height $1/2H < h(B) \leq 3/4H$. We can rearrange the items in this area, such that we generate at most $N^2/4 + N/4 + 4$ boxes for tall items and at most $N^2/2 + N/4 + 6$ boxes for sliced vertical items plus one additional box of height $1/4H$ and width $(1 − 4/(N + 4))w(B)$.

The extra box is placed into the gap, which was generated by shifting all boxes with lower border above $3/4H$ exactly $1/4H$ upwards. If the considered box $B$ has a height of at most $1/2H$ there can be just one tall item per vertical line through this box. In [16] this case was studied and the following Lemma is an adaption of what was proven for this scenario.

**Lemma 5 ([16]).** Let $B$ be a box with height $h(B) \leq 1/2H$. We can rearrange the items in this area, such that we generate at most $N/4 + 1$ boxes for tall items and at most $N/4 + 1$ boxes for sliced vertical items.

In summary, the worst case where we generate the most sub boxes is if $h(B) > 3/4H$. The next step is to find a non fractional placement of the vertical items. We use a linear program and its basic solution to show, that we can place the vertical items, such that only few of them are placed fractionally. These fractional placed items have to be placed in a later step, described in the proof of Lemma 10. A Lemma analogously to the following was proven in [16].

**Lemma 6.** Let $X$ be the number of boxes $B_V$ for vertical items and $Y$ be the number of different sizes of vertical items and $\mu W$ the maximal width of a vertical item. There exists a non fractional placement of the vertical items into the boxes and at most $7(Y + X)$ additional boxes each of height $1/4H$ and width $\mu W$. 

16
Proof. We can define a configuration LP with at most \( Y + X \) equations.

\[
\begin{align*}
\sum_{C \in C_B} X_C &= w(B) \quad \forall B \in B_V \\
\sum_{B \in B_V} \sum_{C \in C_B} X_{C \alpha_i, C} &= w_i \quad \forall i = 1, \ldots, Y \\
X_C &\geq 0 \quad \forall B \in B_V, C \in C_B
\end{align*}
\]

\( w_i \) is the total width of the vertical items with the \( i \)th height. This LP can be solved fractionally using at most \( Y + X \) variables. We place the corresponding configurations into the boxes. Afterward, we place the items into the configurations, such that the last item overlaps the configuration border. Each configuration has a height of at most \( H \). We can partition the overlapping items in a configuration into 7 boxes with height \( \frac{1}{4} H \) and width \( \mu W \). (Stack the items in four boxes greedy such that the last item overlaps the box on top. Since the total height of the items is at most \( H \), there are at most three items overlapping. Each of them is placed into their own box.) In these boxes the items can be placed as a whole, since they have a width of at most \( \mu W \). In total, we generate at most \( 7(Y + X) \) boxes. \( \square \)

4 Structure Result

In this section, we use the results proven in Section 3 to prove the new structural result. While Lemmas 7 to 9 are standard techniques to simplify the input instance, we still need to do some work to prove the Structure Lemma 10. The challenge is to place the \( 7(Y + X) \) extra boxes for vertical items. Unlike in the approaches in [25], [8] or [16], we can not place them on the top of the packing since by the shifting we extend it by \( \frac{1}{4} H \). Fortunately by this shifting, we created some free area where the boxes can be placed.

First, we partition the set of items \( \mathcal{I} \), see Figure 6 for an overview. Let \( \delta \leq \varepsilon \) and \( \mu < \delta \) be suitable constants depending on \( \varepsilon \), and OPT be the height of an optimal packing. We define

- \( \mathcal{L} := \{ i \in \mathcal{I} | h(i) > \delta \text{OPT}, w(i) \geq \delta W \} \) as the set of large items,
- \( \mathcal{T} := \{ i \in \mathcal{I} | h(i) \geq (1/4 + \varepsilon/2) \text{OPT}, w(i) < \delta W \} \) as the set of tall items,
- \( \mathcal{V} := \{ i \in \mathcal{I} | \delta \text{OPT} \leq h(i) < (1/4 + \varepsilon/2) \text{OPT}, w(i) \leq \mu W \} \) as the set of vertical items,
- \( \mathcal{M}_V := \{ i \in \mathcal{I} | \varepsilon \text{OPT} \leq h(i) < (1/4 + \varepsilon/2) \text{OPT}, \mu W < w(i) \leq \delta W \} \) as the set of vertical medium items,
- \( \mathcal{H} := \{ i \in \mathcal{I} | h(i) \leq \mu \text{OPT}, \delta W \leq w(i) \} \) as the set of horizontal items,
- \( \mathcal{S} := \{ i \in \mathcal{I} | h(i) \leq \mu \text{OPT}, w(i) \leq \mu W \} \) as the set of small items and
- \( \mathcal{M} := \{ i \in \mathcal{I} | h(i) < \varepsilon \text{OPT}, \mu W < w(i) \leq \delta W \} \cup \{ i \in \mathcal{I} | \mu \text{OPT} < h(i) \leq \delta \text{OPT} \} = \mathcal{I} \setminus (\mathcal{L} \cup \mathcal{T} \cup \mathcal{V} \cup \mathcal{M}_V \cup \mathcal{H} \cup \mathcal{S}) \) as the set of medium sized items.

We want to choose \( \delta \) and \( \mu \) such that the total area of the items in \( \mathcal{M} \) and \( \mathcal{M}_V \) is small. The following Lemma states that we can find suitable values for \( \delta \) and \( \mu \).

**Lemma 7.** Let be \( y \in \mathbb{N} \). Consider the sequence \( \sigma_0 = \varepsilon^y, \sigma_{i+1} = \sigma_i^2 \varepsilon^y \). There is a value \( i \in \{0, \ldots, 1/\varepsilon^y - 1\} \) such that the total area of the items in \( \mathcal{M} \cup \mathcal{M}_V \) is at most \( \varepsilon^y \text{OPT} \), when we define \( \delta := \sigma_i \) and \( \mu := \sigma_{i+1} \).
Figure 6: Partition of the items. Each item is represented by a dot in this plane. The x-coordinate represents its width while the y-coordinate represents its height.

Proof. This can be proven by the pigeon hole principle. The sequence $\sigma$ and the corresponding choice of $\delta$ and $\mu$ partitions the set of items into $1/\varepsilon$ sets. Since the total area of all items is at most $WOPT$ one of the sets must have an area which is at most $\varepsilon W$. For this application it is sufficient to choose $y = 16$. Since each item in $M_V$ has a height of at least $\varepsilon OPT$ and width of at least $\delta^2 \varepsilon W$ it holds that $|M_V| \leq \varepsilon W / (\delta^2 \varepsilon + W) = 1/\delta^2 \varepsilon$. Furthermore, note that $\sigma_i = \varepsilon^{(1+1)^i}$ and therefore $\delta \geq \sigma_{1/\varepsilon} \geq \varepsilon^k$, where $k := y(2^{1/\varepsilon^y}) = 2^{1/\varepsilon^y} + 4$.

In the next step we round the heights of the items in $L \cup T \cup V \cup M_V$. To do this we use the same rounding strategy as in [16].

Lemma 8 ([15]). Let be $\delta = \varepsilon^k$ for some value $k \in \mathbb{N}$. At loss of a factor $(1 + 2 \varepsilon)$ in the approximation ratio, we can ensure that each item $i$ with height $\varepsilon^{l-1} OPT \geq h(i) \geq \varepsilon^l OPT$ for some $l \in \mathbb{N} \leq k$ has height $k \varepsilon^{l+1} OPT$ for a value $k_i \in \{1/\varepsilon, \ldots, 1/\varepsilon^2 - 1\}$. Furthermore, the items upper and lower border can be placed at multiples of $\varepsilon^{l+1} OPT$.

After this rounding, there are at most $k/\varepsilon^2 \leq 1/\delta \varepsilon$ different heights in $V$ and at most $1/\varepsilon^2$ different heights in $T$. Furthermore, each item in $T$ has its lower border at multiples of $\varepsilon^2 OPT$. Therefore, the arithmetic grid has $N + 1 := (1 + 2 \varepsilon)OPT / \varepsilon^2 OPT + 1 = (1 + 2 \varepsilon) / \varepsilon^2 + 1$ grid lines. We define $H := (1 + 2 \varepsilon)OPT$. Notice that each tall item is taller than $1/4 H = (1/4 + \varepsilon/2)OPT$ and, thus, we can use Lemmas 3 to 5.

In the next step we want to partition the packing area into a constant number of boxes. We have to partition the packing area a little different than in [16] since the items in $M_V$ can not be placed in a later step. We will handle these items the same as the large items $L$. The proof of the following Lemma works analogously as the prove to the similar Lemma in [16].

Lemma 9. We can partition a rounded optimal packing into at most $8(1 + 1 \varepsilon)/\delta^2 \varepsilon$ boxes such that the following conditions hold:

- There are $|L| + |M_V|$ boxes each containing exactly one item from the set $L \cup M_V$ and all items from this set are contained in these boxes.

- There are at most $(1 + 2 \varepsilon)/\delta^2 \varepsilon - |L|$ boxes containing all horizontal items $H$. The horizontal items can overlap horizontal box borders, but never vertical box borders. We call the set of these boxes $B_H$. 

18
Lemma 10. (Structure Lemma) By extending the packing area to $T \cup V$, the set of these items is called the set of boxes for horizontal and small items, which we need by Lemma 6 to place the vertical items non-fractionally. In the final step, we consider the containers for vertical items we constructed and find a place for the resulting set of additional containers, to reorder the items inside the boxes for tall and vertical items. Afterward, we analyze the number of boxes containing all items in $T \cup V$. The items can overlap vertical box borders, but never horizontal box borders. We call the set of these boxes $B_{T \cup V}$.

- The medium sized and small items $M$ and $S$ are completely contained in the boxes $B_H$ and $B_{T \cup V}$ and can overlap them at any box border.

- The number of different vertical lines at box borders in the packing area is bounded by $2((1 + 2\varepsilon)/\delta^2 \varepsilon^2 + |M_V|)$.

- The lower border of each box is positioned at a multiple of $\varepsilon \delta \text{OPT}$.

We use the partitioning form Lemma 9 to prove the following structure result.

Lemma 10. (Structure Lemma) By extending the packing area to $(5/4 + 6\varepsilon)\text{OPTW}$ each rounded optimal packing can be rearranged and partitioned into $O(1/\delta^3 \varepsilon^5)$ boxes with the following properties:

- There are $|L| + |M_V| = O(1/\delta^2 \varepsilon)$ boxes, each containing exactly one item from the set $L \cup M_V$ and all items from this set are contained in these boxes.

- There are at most $(1 + 2\varepsilon)/\delta^2 \varepsilon^2 - |L|$ boxes containing all horizontal items $H$. The horizontal items can overlap horizontal box borders, but never vertical box borders.

- There are at most $O(1/\delta^2 \varepsilon^5)$ boxes containing tall items, such that each tall item $t$ is contained in a box with rounded height $h(t)$.

- There are at most $O(1/\delta^3 \varepsilon^5)$ boxes containing vertical items, such that each vertical item $v$ is contained in a box with rounded height $h(v)$.

- There are at most $14 \cdot 22 \cdot 24/\delta^2 \varepsilon^5 \leq 1/\delta^2 \varepsilon^8$ boxes for small items with total area at least $\text{AREA}(S)$.

- The lower border of each box is positioned at a multiple of $\varepsilon \delta \text{OPT}$.

Proof. In the following, we give a short overview of the steps in this proof. We start with the partition from Lemma 9. The main task in this proof is to find a place for the extra boxes for vertical items, which we need to place them integrally. For this purpose, we consider the widest tall items intersecting the horizontal line $1/2H$ and fix their position. We aim to place the extra boxes on top of them if the total width of these items is large enough. Otherwise, we know that all the tall items intersecting this line are very thin and we can find a way to place the extra boxes inside the boxes with height at least $3/4H$. In the next step, we provide the condition assumed in Section 3. First, we ensure that no box $B$ with height at least $3/4H$ is intersected at its border at the horizontal line $S(B) + H' - H/4$ by a tall item, by introducing at most two further boxes per box $B$. After that, we shift up the boxes which have their lower border above $3/4H$ by $1/4H + \varepsilon$ to ensure that above each box of height $3/4H$ there is a gap of at least $1/4H + \varepsilon$ and make it possible to enlarge the box. Last, we ensure that each box of height larger than $1/2H$ starts or ends at multiples of $\varepsilon^2 \text{OPT}$. When all these properties are fulfilled, we can apply Lemmas 3, 4 and 5 to reorder the items inside the boxes for tall and vertical items. Afterward, we analyze the number of containers for vertical items we constructed and find a place for the resulting set of additional containers, which we need by Lemma 6 to place the vertical items non-fractional. In the final step, we consider the boxes for horizontal and small items.

Wide tall items. First, we look at the $1/\delta^2 \varepsilon$ widest tall items crossing the horizontal line $1/2H$. We call the set of these items $T_{1/2H}$. Each of these items defines a new unmovable item. It splits the box containing it into three parts: the part left of this item, the part right of this item and the part containing this item. The parts at the left and the right will be reordered as any other box, while the part containing
this item is reordered differently. The item itself is not moved, while the part above and below has a height of less than \( \frac{1}{2}H \). These parts can be reordered by Lemma 5 such that they create at most \( 2(L/4 + 1) \) sub boxes for tall and vertical items total. This number is smaller than the number of boxes for one box of height larger than \( \frac{3}{4}H \). Therefore, we can count this part as one box without making any error and assume that we add at most \( 2/\delta^2\varepsilon \) boxes. After this step, the total number of boxes containing both, tall and vertical items, is at most \( 3((1 + 2\varepsilon)/\delta^2\varepsilon + |M_V|) + 2/\delta^2\varepsilon \leq 8(1 + \varepsilon)/\delta^2\varepsilon \). Furthermore, the number of vertical lines at box borders through the strip is bounded by \( 2((1 + 2\varepsilon)/\delta^2\varepsilon + |M_V|) + 2/\delta^2\varepsilon \leq 6/\delta^2\varepsilon \).

Providing the conditions assumed in Section 3. In the next step, we look at each box \( B \) with height at least \( \frac{3}{4}H \), see Figure 7. Remember that in Lemma 3, we had assumed that no tall item overlaps \( B \)'s left or right box border at \( S(B) + H' - \frac{1}{4}H \). We will establish this property by introducing two boxes for tall and vertical items of height less than \( \frac{3}{4}H \). Assume there is a tall item \( t \) overlapping the left box border at \( S(B) + H' - \frac{1}{4}H \), see Figure 7. We draw a vertical line \( L \) at the right border of \( t \) inside our box. Tall items crossed by \( L \) represent new unmovable items. Obviously, \( L \) is not intersected by a tall item at height \( S(B) + H' - \frac{1}{4}H \). We do the same on the right side of the box. Consider the rectangular area between the left border of the box \( B \) and \( L \) bounded on top by \( t \). This area builds a new box for vertical and tall items with height less than \( \frac{3}{4}H \) and will later be reordered accordingly. These box can be reordered by Lemma 5 such that they create at most \( 2(L/4 + 1) \) sub boxes for tall and vertical items total. The rectangular area above \( t \) between these vertical lines builds pseudo item containing vertical items. This box for vertical items together with the box unmovable item and the subboxes generated by the reordering of the box of height less than \( \frac{3}{4}H \) are less boxes than a reordering of a box with height larger than \( \frac{3}{4}H \) provokes. Therefore we count the left part and the right part as one box each without counting to less boxes. In this step, we create at most \( 2 \cdot 8(1 + \varepsilon)/\delta^2\varepsilon \) extra boxes for tall and vertical items and the total number of boxes containing both, tall and vertical items, is bounded by \( 3 \cdot 8(1 + \varepsilon)/\delta^2\varepsilon \leq 24(1 + \varepsilon)/\delta^2\varepsilon \). This number will not increase in the following steps and we define \( N_B := 24(1 + \varepsilon)/\delta^2\varepsilon \leq 1/\delta^2\varepsilon \), if \( \varepsilon \leq 1/25 \). Furthermore, the number of vertical lines at box border through the strip is bounded by \( 6(1 + \varepsilon)/\delta^2\varepsilon + 2 \cdot 8(1 + \varepsilon)/\delta^2\varepsilon \leq 22(1 + \varepsilon)/\delta^2\varepsilon =: N_L \).

Now, we draw a horizontal line at \( \frac{3}{4}H \) through the strip and shift each box with lower border above or at this line exactly \( (1/4 + 3\varepsilon/2)OPT = \frac{1}{4}H + \varepsilon OPT \) upwards. Note that no tall item is shifted, since they start before \( \frac{3}{4}H \) and therefore their boxes do too. After this shift, on top of each box with height

---

**Figure 7:** Excluding overlappings at \( H' - \frac{1}{4}H \)
at least \( \frac{3}{4}H \), there is a gap of height \( \frac{1}{4}H + \varepsilon \text{OPT} \). Notice that we add an extra \( \varepsilon \text{OPT} \) to the height. Therefore in the later reordering of boxes with height at least \( \frac{3}{4}H \), we have to shift the items crossing the line \( H' - \frac{1}{4}H \) exactly \( \frac{1}{4}H + \varepsilon \text{OPT} \) upwards and not just \( \frac{1}{4}H \) as in the proof of Lemma 3. Consider a box \( B \) of height larger than \( \frac{1}{2}H \) and at most \( \frac{3}{4}H \). To rearrange the items in the box \( B \), we need an extra box with height \( \frac{1}{4}H \) and width \((1 - 4/(1/\varepsilon^2 + 4))w(B) \leq (1 - \varepsilon^2)w(B) \) by Lemma 4. Due to the shifting, somewhere above this box, there is free area of height \( \frac{1}{4}H + \varepsilon \text{OPT} \) and width \( w(B) \), which is possible divided into several vertical slices. Let us look at the free area above all the boxes with height \( \frac{3}{4}H \) and \( \frac{1}{2}H \). This free area is scattered into at most \( N_L + 1 \) vertical pieces since there are at most \( N_L \) vertical lines at box borders. We allocate this free area above the boxes as contiguously as possible. For each piece of the free area we use, we introduce one box for vertical items (at most \( N_L + 1 \)). Let \( W_{1/2} \) be the total width of boxes with height larger than \( \frac{1}{2}H \) and at most \( \frac{3}{4}H \). The total width of the free area above this boxes we use to place the extra boxes is bounded by \((1 - \varepsilon^2)W_{1/2} \) and we have a total width of at least \( \varepsilon^2W_{1/2} \) to place the extra boxes.

For the reordering in Lemmas 3 and 4 we had assumed, that each box with height larger than \( \frac{1}{2}H \) starts and ends at grid points. In this step, we generate this property. As grid lines we had defined all multiples of \( \varepsilon \text{OPT} \). Let \( B \) be a box with height larger than \( \frac{1}{2}H \). Look at the horizontal line \( l \) at the smallest multiple of \( \varepsilon \text{OPT} \) in this box. The distance between \( l \) and the bottom border is smaller than \( \varepsilon \text{OPT} \). In the box \( B \), we remove all the vertical items below and each item cut by \( l \) and position them in an extra box above the packing area \( \frac{3}{4}H \). Since each item with height larger than \( \varepsilon \text{OPT} \) starts and ends at multiples of \( \varepsilon \text{OPT} \), the items cut by \( l \) have a height of at most \( \varepsilon \text{OPT} \). We do the same on top of this box and for each other box. We create above \( \frac{3}{4}H + \varepsilon \text{OPT} \) a box with height \( 2(\varepsilon + \varepsilon^2)\varepsilon \text{OPT} \) and width \( W \). All the removed items fit inside this box since there is at most one of the boxes with height larger than \( \frac{1}{2}H \) per vertical line through the strip.

Next, we reorder the boxes as described in Lemmas 3 to 5. We create at most \( 2N^2 + 15N/4 + 8 \leq 2(1 + 18\varepsilon)/\varepsilon^4 \) sub boxes for tall and at most \( 4N^3 + 31N/4 + 4 \leq 4(1 + 18\varepsilon)/\varepsilon^4 \) sub boxes for vertical items per box for tall and vertical items; (remember \( N = (1 + 2/\varepsilon)^2 \) and \( \varepsilon \leq 1/25 \). To this point, we generate at most \( 24(1 + \varepsilon)/\delta^2 \varepsilon \cdot 2(1 + 18\varepsilon)/\varepsilon^4 = 24 \cdot 3/\delta^2 \varepsilon^5 \) boxes for tall items. Consider the boxes with height larger than \( \frac{3}{4}H \). In these boxes, we draw a vertical line at each sub box’s left border. If a box for vertical items is intersected by a vertical line, we split the box at this line. Each line intersects at most 3 boxes for vertical items. After this splitting of boxes, the total number of sub boxes for vertical items in a box of height larger than \( \frac{3}{4}H \) is bounded by \( 4(1 + 18\varepsilon)/\varepsilon^4 + 3(4(1 + 18\varepsilon)/\varepsilon^4 + 2(1 + 18\varepsilon)/\varepsilon^4) = 22(1 + 18\varepsilon)/\varepsilon^4 \). The area between two consecutive lines defines a strip, where the height of all the intersected boxes does not change. We have at most \( 4(1 + 18\varepsilon)/\varepsilon^4 + 2(1 + 18\varepsilon)/\varepsilon^4 = 6(1 + 18\varepsilon)/\varepsilon^4 \) of these strips per box of height larger than \( \frac{3}{4}H \). Therefore, the total number of strips is bounded by \( N_S := 6(1 + 18\varepsilon)/\varepsilon^4 \cdot 24(1 + \varepsilon)/\delta^2 \varepsilon \leq 24 \cdot 13/\delta^2 \varepsilon^5 \leq 1/\delta^2 \varepsilon^7 \), while the total number of boxes for vertical items is bounded by \( 22(1 + 18\varepsilon)/\varepsilon^4 \cdot 24(1 + \varepsilon)/\delta^2 \varepsilon + N_L + 1 \leq 22 \cdot 24(1 + 19\varepsilon + 18\varepsilon^2)/\delta^2 \varepsilon^5 \).

**Placing the extra boxes.** By Lemma 6, we need at most \( 7(Y + X) \) additional boxes with height \( \frac{1}{4}H \) and width \( \mu W \) to place the vertical items non fractionally into the boxes. We call the set of these boxes \( B_{\mu W} \). We can bound the variables in the following way. There are at most \( X \leq 22 \cdot 24(1 + 19\varepsilon + 18\varepsilon^2)/\delta^2 \varepsilon^5 \) boxes for vertical items and at most \( Y \leq 1/\varepsilon \) different heights of the items. Therefore, we need at most \( 7(22 \cdot 24(1 + 19\varepsilon + 18\varepsilon^2)/\delta^2 \varepsilon^5 + e^4 \delta /\delta^2 \varepsilon^5) \leq 13 \cdot 22 \cdot 24 \cdot 1/\delta^2 \varepsilon^5 \) additional boxes \( B_{\mu W} \), if \( \varepsilon \leq 1/25 \).

We have to place the additional boxes inside the packing area \( W \cdot 5/4H \). In the following steps, we will prove that it is possible to place them, by considering three possibilities. Consider again the vertical lines at the box borders. These \( N_L \) lines generate at most \( N_L + 1 \) strips. Notice that the free area in the strips is just scattered in strips containing boxes with height larger than \( \frac{3}{4}H \). Let \( W_T \) be the total width of the strips containing items from \( T_{1/2H} \). \( W_H \) be the total width of the strips containing boxes with height at least \( \frac{3}{4}H \) and \( W_R \) be the total width of all other strips. In total we have \( W_T + W_H + W_R = W \). We can assume \( N_B \leq 1/\delta^2 \varepsilon^2 \), \( N_S \leq 1/\delta^2 \varepsilon^7 \), \( N_F \leq 1/\delta^2 \varepsilon^8 \) and \( N_L \leq 1/\delta^2 \varepsilon^2 \), if \( \varepsilon \leq 1/24 \).
Consider the strips without boxes of height $3/4H$ or the items in $T_{1/2H}$. These strips can contain boxes with height larger than $1/2H$. Therefore, we have free area with total width at least $\varepsilon^2 W_R$ in these strips.

**Claim.** If $W_R \geq 2\varepsilon^6 W$, we can place the $N_F$ boxes $B_{\mu W}$ into these areas.

The now considered Strips contain boxes with height larger than $1/2H$ and less than $3/4H$. Therefore the free area in this strips can be used partially by the extra boxes for vertical items for these boxes. Nevertheless, these strips contain free area with width at least $\varepsilon^2 W_R$ we can use to place the now considered extra boxes. In each of these at most $(N_L + 1)$ strips the free area is contiguous. However, we have to calculate a small error we could make, because each of the boxes to place has a width of $\varepsilon\mu W$ and therefore it can happen that in each strip we have a residual width of $\mu W - 1$ where we can not place a box. On the positive side, we can use an area with total width at least $\varepsilon^2 W_R - (N_L + 1)\mu W$ to place the boxes in $B_{\mu W}$. Therefore if $\varepsilon^2 W_R - (N_L + 1)\mu W \geq N_F \mu W$, we can place all the boxes. Using $\mu := 1/\delta^2 \varepsilon^{16}$, it holds that

$$N_F \mu W + (N_L + 1)\mu W = \delta^2 \varepsilon^{16} W(1/\delta^2 \varepsilon^8 + 2/\delta^2 \varepsilon^2 + 1) \leq 2\varepsilon^{16} W \varepsilon^8 \leq 2\varepsilon^8 W.$$ 

Therefore, if $W_R \geq 2\varepsilon^6 W$, it holds that $\varepsilon^2 W_R - (N_L + 1)\mu W \geq N_F \mu W$ and we can place all the boxes $B_{\mu W}$, which concludes the claim.

**Claim.** If $W_T \geq 2\varepsilon^8 W$, we can place the $N_F$ boxes $B_{\mu W}$ in the strips containing the items in $T_{1/2H}$.

There are at most $N_L + 1$ strips containing parts of the items in $T_{1/2H}$. In these strips the free area is contiguous and can be fully used, since these strips do not contain boxes with height larger than $1/2H$. Each box in $B_{\mu W}$ has a width of exactly $\mu W$. Hence, in each strip there is an area with width at most $\mu W$ which we can not use to place the boxes. Therefore, if $W_T - \mu WN_L \geq N_F \mu W$ we can place all the $N_F$ boxes into these strips. Using $\mu := 1/\delta^2 \varepsilon^{16}$, it holds that

$$N_F \mu W + \mu WN_L = \delta^2 \varepsilon^{16} W(1/\delta^2 \varepsilon^8 + 1/\delta^2 \varepsilon^2) \leq 2\varepsilon^8 W$$

Therefore, if $W_T \geq 2\varepsilon^8 W$, it holds that $W_T - \mu WN_L \geq N_F \mu W$ and we can place all the boxes $B_{\mu W}$, which concludes the claim.

**Claim.** If $W_T < 2\varepsilon^8 W$ as well as $W_H < 2\varepsilon^6 W$, we can place all the boxes for vertical items inside the boxes of height at least $3/4H$.

In this case it holds that $W_H > (1 - 3\varepsilon^6)W \geq 5\varepsilon W$. Furthermore, each tall item not in $T_{1/2H}$ crossing $1/2H$ has a width of at most $2\delta^2 \varepsilon^9 W := w_{\max}$. After the reordering in the boxes, there are at most $N_S$ strips in the boxes total. We consider the added free area in each strip. We are interested in the total height of this free area, which might be not connected. By the shifting step we have added to these strips a total area of $W_H(1/4H + \varepsilon OPT)$. Let $\hat{W}_H$ be the total width of the strips containing free area with total height less than $1/4H$ and let $\hat{W}_H$ be the total width of strips containing free area with height larger than $1/4H$. In each strip the total free area can have a height of at most $3/4H + \varepsilon OPT$, since on top and on the bottom there are always boxes with height at least $1/4H$ or there is a box with height at least $3/4H$ on the bottom. It holds that $\hat{W}_H + \hat{W}_H = W_H$ and

$$W_H(1/4H + \varepsilon OPT) \geq W_H(1/4H + \varepsilon OPT)$$

$$\hat{W}_H(1/2H + \varepsilon OPT) \geq W_H(\varepsilon OPT)$$

$$\hat{W}_H(1/2 + \varepsilon/(1 + 2\varepsilon)) \geq W_H(\varepsilon/(1 + 2\varepsilon))$$

$$\hat{W}_H(1 + 4\varepsilon)/2(1 + 2\varepsilon) \geq W_H(2\varepsilon/2(1 + 2\varepsilon))$$

$$\hat{W}_H \geq W_H(2\varepsilon/(1 + 4\varepsilon)) \geq \varepsilon W_H$$
Therefore, strips with total width at least \( \varepsilon W_H \) contain free area with total height at least \( \frac{1}{4} H \). The free area in this strips can be scattered. We will fuse this free area by shifting the boxes for vertical or tall items. Notice that we can shift the boxes for vertical items in each strip freely up and down, since their box borders are at the strip borders by construction. This is different for the sub boxes for tall items, which can be positioned between \( \frac{1}{2} H' \) and \( H' - \frac{1}{4} H \). These sub boxes possibly contain tall items overlapping the strip’s borders. Remember that each tall item in this strip has a width of at most \( w_{\max} = 2 \varepsilon^9 \delta^2 W \). Hence, in each strip with width larger than \( 2 w_{\max} = 4 \varepsilon^9 \delta^2 W \), we can shift the middle part of these sub boxes such that the free area is connected. We do not shift the sub boxes touching the bottom or the top of the box. In each strip, there is an area with width at most \( \mu W \) which we can not use to place the boxes. Therefore, we can place all boxes for previously fractional vertical items, if \( \varepsilon W_H - 2 w_{\max} N_S - \mu W N_S \geq \mu W N_F \). It holds that

\[
2 w_{\max} N_S + \mu W N_S + \mu W N_F \leq (4 \varepsilon^9 \delta^2 W) / \delta^2 \varepsilon^7 + \delta^2 \varepsilon^{16} W (1 / \delta^2 \varepsilon^7 + 1 / \delta^2 \varepsilon^8) \leq 5 \varepsilon^2 W
\]

Therefore, if \( W_R \geq 5 \varepsilon W \), it holds that \( \varepsilon W_H - 2 w_{\max} N_S - \mu W N_S \geq \mu W N_F \) and we can place all boxes in this case, which concludes the claim.

In this step, we create at most \( 2 N_S \) new boxes for tall items and no new box for vertical items. The boxes for tall items already do contain just tall items with the same height. Therefore, we introduce at most \( 24 \cdot 3 / \delta^2 \varepsilon^5 + 2 N_S \leq (24 \cdot 29) / \delta^2 \varepsilon^5 \) boxes for tall items.

Furthermore, we have at most \( 8(Y + X) \leq 8 \cdot 22 \cdot 24(1 + 19 \varepsilon + 18 \varepsilon^2) / \delta^2 \varepsilon^5 \) boxes and configurations for vertical items. Each configuration can contain at most \((1 + 2 \varepsilon) / \delta \) items. Therefore, we create at most \( 8 \cdot 22 \cdot 20(1 + 2 \varepsilon)(1 + 19 \varepsilon + 18 \varepsilon^2) / \delta^2 \varepsilon^5 \leq 16 \cdot 22 \cdot 20 / \delta^2 \varepsilon^5 \) boxes for vertical items, such that each box \( B \) contains just items with height \( h(B) \).

**Small items.** The configurations for vertical and horizontal items cover exactly the area of the previously placed items. Therefore in total, they contain enough free area for the small items. In each configuration, we have one box for small items. There are at most \((1 + 2 \varepsilon + 1) / \delta^2 \varepsilon \) configurations for horizontal items and at most \( 13 \cdot 22 \cdot 24 / \delta^2 \varepsilon^5 \) configurations for vertical items. Hence, we generate at most \( 14 \cdot 22 \cdot 24 / \delta^2 \varepsilon^5 \leq 1 / \delta^2 \varepsilon^8 \) boxes for small items with a total area of \( \text{AREA}(S) \), if \( \varepsilon \leq 1 / 24 \).

**Packing height.** Let us recapitulate what we added to the packing height during this process. We started with a packing of height \( \text{OPT} \). After the rounding of the items with height larger than \( \delta \), we received a packing with height \((1 + 2 \varepsilon) \text{OPT} \). With the shifting at the horizontal line \( 3(1 + 2 \varepsilon) \text{OPT} / 4 \) we added \((1 + 2 \varepsilon) \text{OPT} / 4 + \varepsilon \text{OPT} \) to the packing height. Finally, we shifted some vertical items to ensure that the boxes with height taller than \( \frac{1}{2} H \) start and end at multiples of \( \varepsilon^2 \text{OPT} \). This added further \( 2(\varepsilon + \varepsilon^2) \text{OPT} \) to the packing height. In total, the structured packing has a height of at most \((\frac{5}{4} + \frac{11}{10} \varepsilon + 2 \varepsilon^2) \text{OPT} \) to the packing height. We now have a packing height of at most \((\frac{5}{4} + 6 \varepsilon) \text{OPT} \).

The last step is to see that it is possible to place the horizontal and small items, without enlarging the packing height to much.

**Lemma 11.** It is possible to place the horizontal items \( H \) into the at most \((1 + 2 \varepsilon) / \delta^2 \varepsilon \) boxes for horizontal items and one additional box of height \( \varepsilon \text{OPT} \) and width \( W \) in at most \( n \log(n) + 1 / \delta^{O(1/\delta^2 \varepsilon)} \) operations.

**Proof.** The first step is to round the horizontal items. We stack horizontal items on top of each other ordered by their width, such that the widest item is positioned at the bottom. This stack has a height of at most \( \text{OPT} / \delta \), since each item has a width of at least \( \delta W \) and their total area is bounded by \( \text{OPT} \cdot W \). We group the items in the stack to at most \( 1 / \delta^2 \varepsilon \) groups, each of height \( \delta^2 \varepsilon \text{OPT} / \delta = \delta \varepsilon \text{OPT} \) and round the items in the groups to the widest width occurring in the group. By this we generate at most \( 1 / \delta^2 \varepsilon \) different sizes. The rounded horizontal items can be placed fractionally into the non rounded items of the group containing the next larger items. The group containing the widest items has to be placed on top...
if the set. Therefore, the total height of items we put on the top of the packing has a height of at most \( \delta \varepsilon \text{OPT} + \mu \text{OPT} \), since the item overlapping the rounding border has height of at most \( \mu \text{OPT} \).

Now we place the rounded horizontal items into the boxes. The following configuration LP is solvable since the rounded horizontal items fit fractionally into the boxes \( B_H \).

\[
\sum_{C \in C_B} X_C = \varepsilon \delta \text{OPT} \quad \forall B \in B_H
\]
\[
\sum_{B \in B_H} \sum_{C \in C_B} X_C a_{i,C} = h_i \quad \forall i = 1, \ldots, 1/\delta^2 \varepsilon - 1
\]
\[
X_C \geq 0 \quad \forall B \in B_H, C \in C_B
\]

This LP can be solved using at most \(|B_H| + 1/\delta^2 \varepsilon \) variables in at most \( 1/\delta^O(1/\delta^2 \varepsilon) \) operations. We place the corresponding configurations into the corresponding boxes. We place the items into the configurations, such that the last item overlaps the configuration border. In the next step, we remove these overlapping items and place them on top of the box. These items have a height of at most \( \mu \text{OPT} \). Since a basic solution has at most \((1 + 2\varepsilon + 1)/\delta^2 \varepsilon \) configurations, we add at most \( \mu \text{OPT}(2 + 2\varepsilon)/\delta^2 \varepsilon \leq (2 + 2\varepsilon) \varepsilon^{15} \text{OPT} \) to the packing height. In total we add at most \( \delta \varepsilon \text{OPT} + \mu \text{OPT} + \mu \text{OPT}(2 + 2\varepsilon)/\delta^2 \varepsilon \leq \varepsilon \text{OPT} \) to the packing height.

**Lemma 12.** It is possible to place the small items \( S \) into the at most \( 1/\delta^2 \varepsilon^8 \) boxes for small items and one additional box of height \( 2\varepsilon^7 \text{OPT} \) and width \( W \).

**Proof.** We place the items with the NFDH algorithm into the boxes. First, we discard any box with height less than \( \mu \text{OPT} \) or width less than \( \mu W \). The total area of each discarded box is at most \( \mu W \text{OPT} \). Let us consider a box \( B \) with height and width larger than \( \mu \text{OPT} \) or \( \mu W \) respectively. In each shelf we use for the NFDH-Algorithm, we can not use a total width of at most \( \mu W \) to place the items. Furthermore, the last shelf has a distance of at most \( \mu \text{OPT} \) to the upper border of the box. Additionally, the free area between the shelves has a total area of at most \( \mu \text{OPT} \cdot w(B) \). Therefore, the total free area in \( B \) is at most \( \mu W \cdot h(B) + 2\mu \text{OPT} \cdot w(B) \leq 3\mu W \text{OPT} \). As a result, the total area of items that could not be placed inside the boxes is at most \( 3\mu W \text{OPT} \cdot \mu W \text{OPT} / \delta^2 \varepsilon^8 \leq \varepsilon^7 \mu W \text{OPT} \). These items can be place with Steinberg’s algorithm [29] into a box with width \( W \) and height \( 2\varepsilon^7 \text{OPT} \). \( \square \)

## 5 Algorithms

In this section, we describe the three algorithms for Strip Packing without rotations, Strip Packing with rotations and Scheduling contiguous moldable Jobs. Each optimal solution of these three problems can be rearranged, such that the structure looks like the structure in Lemma 10. The algorithms all work roughly the same. First, we determine an upper bound for the approximation. Afterward, we use a binary search framework to find a \((5/4 + \varepsilon)\) approximation. The routine called by the framework guesses the structure of the packing and tests with a dynamic program if the guess is feasible.

### 5.1 Strip Packing without Rotations

Given a value \( \varepsilon \in (0, 1] \) and an instance \( I \), we define \( \varepsilon' \) as the largest value such that \( 1/\varepsilon' \in \mathbb{N}, \varepsilon' \leq \varepsilon/12 \) and \( \varepsilon' \leq 1/25 \). Next, we estimate the packing height \( \text{OPT} \) by using that Steinberg’s algorithm [29] can place all items into a strip with height of at most \( 2\text{AREA}(I)/W \). Therefore, we know that \( \text{OPT} \in [\text{AREA}(I)/W, 2\text{AREA}(I)/W] \).

Afterward, we use a dynamic search framework, to find a packing with height at most \((5/4 + O(\varepsilon'))(1 + \varepsilon')T\) with \( \text{OPT} \in [T, (1 + \varepsilon')T] \). For this purpose, we assume \( \text{OPT} = (1 + \varepsilon')T \) for a given value \( T \). Next,
we compute the values $\delta$ and $\mu$ with the properties from Lemma 7 while assuming $\text{OPT} = (1 + \varepsilon')T$. Knowing these values, we partition and round the items accordingly, see Lemma 8 and Lemma 11.

Afterward, we guess the structure of an approximative solution via Lemma 10 with packing height of at most $(\frac{5}{4} + 6\varepsilon')(1 + \varepsilon')T$. For each of the at most $O(1/\delta^2\varepsilon^5)$ boxes, we guess the lower left corner and the upper right corner. For each box there are at most $O((W/\delta\varepsilon)^2)$ possibilities to guess these positions since the x-coordinates are in $\{0, \ldots, W - 1\}$, and the y-coordinates are in $\{0, \ldots, O(1)/\delta\varepsilon\}$. Therefore, there are at most $(W/\delta\varepsilon)^O(1/\delta^2\varepsilon^5)$ possible guessing steps. A guessed structure is feasible if we can place the items into the corresponding boxes.

For each of the guesses, we test if we can place the items in $V \cup T$ into the guessed boxes by using the following dynamic program: For each rounded height $\tilde{h}$ we generate a vector $(w_{\tilde{h},1}, \ldots, w_{\tilde{h},k_h})$. It represents the $k_h$ boxes for items $i \in V \cup T$ with height $h(i) = \tilde{h}$. Each entry is bounded by the width of the corresponding box. For each rounded height $\tilde{h}$ the program enumerates all items $i \in V \cup T$ with height $h(i) = \tilde{h}$. We start with the vector $(w_{\tilde{h},1} = 0, \ldots, w_{\tilde{h},k_h} = 0)$. For each item $i \in T \cup V$ with $h(i) = \tilde{h}$, we make $k_h$ copies of each so far generated vector $(w_{\tilde{h},1}, \ldots, w_{\tilde{h},k_h})$ and add the value $w(i)$ to a different component in each copy. If we enlarge one component above its maximal value, we discard this vector. The guess of boxes for items with height $\tilde{h}$ was feasible if after enumerating all the items with height $\tilde{h}$ there is still a valid vector. The width of each box for tall and vertical items $T \cup V$ is bounded by $W$. Therefore, we enumerate at most $W^{O(k)}$ vectors, where $k$ is the total number of boxes. Therefore, the dynamic program has a running time of at most $n \cdot W^{O(1/\delta^2\varepsilon^5)}$.

If it is possible to place the items in $L \cup V \cup T \cup M_V$, we place the tall items with the algorithm described in the proof of Lemma 11 and the small items into their boxes with the NFHD algorithm. By Lemma 12, placing the small items adds at most at most further $2\varepsilon^7(1 + \varepsilon')T$ to the packing height while by Lemma 11 placing the horizontal items adds at most $\varepsilon(1 + \varepsilon')T$ to the packing height. Afterward, we place the items in $M$ on top of the packing. These items add at most $2\varepsilon'(1 + \varepsilon')T$ to the packing height, since each item in $M$ has a height of at most $\varepsilon'(1 + \varepsilon')T$ and the total area of these items is at most $\varepsilon'^{16}(1 + \varepsilon')T$.

If we can not find a packing for any of the guesses, the value $T$ was to low and we adjust the lower bound for the binary search. On the other hand if we find a solution, we get a solution with height at most $(5/4 + 10\varepsilon')(1 + \varepsilon')T \leq (5/4 + 12\varepsilon')T$ and adjust the upper bound for the binary search. When the binary search terminates, we have found a packing with height at most $(5/4 + 12\varepsilon')T$ for $\text{OPT} \in [T, (1 + \varepsilon')T]$. Since $\varepsilon' \leq \varepsilon/12$ and $T \leq \text{OPT} \leq (1 + \varepsilon')T$ it holds that $(5/4 + 12\varepsilon')T \leq (5/4 + \varepsilon)\text{OPT}$. Using $\delta \leq \varepsilon^{2/\epsilon^{16} + 4}$, the running time of the algorithm is bounded by $O(n \log(n)) + O(\log(1/\varepsilon'))(W/\delta\varepsilon)^O(1/\delta^2\varepsilon^5)(W^{O(1/\delta^2\varepsilon^5)} + 1/\delta^{O(1/\delta^2\varepsilon^5)}) + \leq O(n \log(n)) + n \cdot W^{1/\varepsilon^{2O(1/\epsilon^{16})}}$, since the items need to be sorted just once in the beginning of the algorithm.

5.2 Strip Packing with Rotations

In this scenario, we can not round the items in advance, since we do not know the rotation in which the items will be placed (0 degrees or 90 degrees). Furthermore, an item $i$ can be contained in the set $M$ if it is rotated and not contained in $M$ otherwise. Hence, even the computation of the values $\delta$ and $\mu$ is not simple.

Similar as in the algorithm without rotations, we use a binary search framework to find the packing. As a new step, we want to round the heights of all items to multiples of $\varepsilon\text{OPT}/n$. In an optimal packing we know in which rotation an item is placed and hence could round the corresponding item height to a multiple of $\varepsilon\text{OPT}/n$. This rounding would extend the packing height by at most $(1 + \varepsilon)\text{OPT}$, since there are at most $n$ items on top of each other. Therefore, we assume $\text{OPT} = (1 + \varepsilon)^2T$ for a given value $T$, such that $\text{OPT} \in [T, (1 + \varepsilon)T]$. 25
In the first step, we guess the values of $\delta$ and $\mu$ since we can not determine them. There are at most $1/\varepsilon^{16}$ possibilities. Knowing these values, we can decide for a given item and its rotation (0 degrees or 90 degrees) in which set $L, V, H, T, M, M_V$ or $S$ this item is contained in the optimal solution.

In the next step, we consider the horizontal items. We want to round the horizontal items similar as in Lemma 11. However since we do not now in advance which items the set $H$ contains, we have to guess the rounded widths: First, we guess the total height $h(H)$. It holds that $h(H) \in \{iz(1+\varepsilon)^2T/n | i = 0, \ldots , n/\varepsilon\}$, since we can assume that each horizontal item has a rounded height, which is a multiple of $\varepsilon(1+\varepsilon)^2T/n$ and the stack of these items has a total height of at most $(1+\varepsilon)^2T/\varepsilon$. Therefore we need at most $n/\varepsilon\delta$ guessing steps. After this guess, we compute the corresponding number of groups in the linear grouping step and guess the at most $1/\varepsilon\delta^2$ items and their rotation (0 degrees or 90 degrees) defining the rounded widths of the groups (at most $(2n)^1/\varepsilon\delta^2$ possibilities, due to rotation). The corresponding items are placed in the last step on top of the packing. These items add at most $\mu(1+\varepsilon)^2T/\varepsilon\delta^2 \leq \varepsilon^{10}(1+\varepsilon)^2T$ to the packing height. Now we can assume in the dynamic program, that each rounded width occurs with a total height of at most $\delta^2\varepsilon'(1+\varepsilon)^2T$.

In the following let $\tilde{h}(i)$ and $\tilde{w}(i)$ be the rounded height and rounded width for an item for a given rotation (0 degrees or 90 degrees). If an item is contained in the set $L \cup V \cup T \cup M_V$, its height is a multiple of $\varepsilon\delta(1+\varepsilon)^2T$, while its width remains original. If the item is in $H \cup M \cup S$, its height is a multiple of $\varepsilon(1+\varepsilon')T/n$ and its width is either its original or one of the at most $1/\varepsilon\delta^2$ guessed width values.

In the next step, we guess the structure from Lemma 10 using a height of at most $(\frac{5}{4}+6\varepsilon')(1+\varepsilon')^2T$. We guess the $O(1/\delta^2\varepsilon)$ large $L$ and medium vertical $M_V$ items and their rotations in the optimal packing and the structure of the boxes in $W^{1/\delta^2\varepsilon^3} \cdot n^{1/\delta^2\varepsilon}$ possibilities. In total we have at most $1/\varepsilon^{16} \cdot W^{1/\delta^2\varepsilon^3} \cdot n^{1/\delta^2\varepsilon} \cdot n/\varepsilon\delta \cdot (2n)^1/\varepsilon\delta^2 \leq (W/\varepsilon\delta)^{O(1/\varepsilon^5\delta^3)} \cdot O(n^{1/\delta^2})$ possible guesses.

Again we check with a dynamic program if the guess is feasible. We introduce the vectors $w_{hi} = (w_{h_{i,1}}, \ldots , w_{h_{i,k_h}})$ and a vector $h = (h_{w_1}, \ldots , h_{w_{1/\varepsilon^2}})$ for each rounded height $h_i$ and each rounded width $w_j$ respectively. Furthermore, we introduce two values $a_s$ and $a_m$. These values represent the total area of the small items $S$ and medium sized items $M$ receptively. In the dynamic program, we consider a sequence of sets $D_i, i \in I$, containing vectors of the form $(h_{w_1}, \ldots , h_{w_{1/\varepsilon^2}}, w_{h_{1/\varepsilon}}, \ldots , w_{h_{1/\varepsilon}}, a_s, a_m)$. $D_0$ contains just the vector filled with zeros. Iterating over the set of items for each item $i \in I$, we determine for both possible rotations the set it would be contained in. For both rotations, we do the following steps to the set $D_{i-1}$.

- If the $i$th item is in $V \cup T$, we make $k_{h(i)}$ copies of each vector in $D_{i-1}$. In each copy, we add the items width $w(i)$ to an other entry of the vector $w_{h(i)}$ and add it to the set $D_i$.

- If the $i$th item is in $H$, we make a copy of each vector in $D_{i-1}$. In each copy, we add its height $\tilde{h}(i)$ to a the entry $h_{\tilde{w}(i)}$ and add it to the set $D_i$.

- If the $i$th item is in $S$, it holds that $\tilde{h}(i) = l_i \varepsilon(1+\varepsilon)T/n$ for some $l_i \in \{1, \ldots , n/\varepsilon\}$. We make a copy of each vector in the set $D_{i-1}$ and add $l_i \cdot w(i)$ to the value $a_s$ in each vector and add it to the set $D_i$.

- If the $i$th item is in $M$, we do the same as in the previous case with the difference that we add the value $l_i \cdot w(i)$ to $a_m$.

If a value in the vector exceeds its boundary, we discard this vector. Furthermore if a specific vector is created a second time, we save it just once.

The values $h_{i,j}$ are bounded by the height of the rounded group normalized by the rounded item height $\delta^2\varepsilon'(1+\varepsilon')(2T/(\varepsilon'(1+\varepsilon')^2T/n)) = \delta^2n$, while the values $w_{i,j}$ are bounded by the guessed width of
the corresponding box. Let $A_s$ be the total area of the boxes for small items. We bound the value $a_s$ by 
$\lceil nA_s/\varepsilon(1 + \varepsilon')T \rceil \in \mathcal{O}(nW)$, since $A_s \leq (5/4 + 6\varepsilon')(1 + \varepsilon)^2TW$. Furthermore, we bound the value $a_m$ by $(1 + \varepsilon')nW\varepsilon^{15}$ since the medium sized items have an total area of at most $\varepsilon^{16}(1 + \varepsilon')^2TW$ and each item has a height, which is a multiple of $\varepsilon'(1 + \varepsilon')T/n$.

Let us analyze the running time of the linear program. The entries $w_{i,j}$ can take at most $W$ different values, the entries $h_{i,j}$ can take at most $\delta^3n$ different values and the entries $a_s$ and $a_m$ can take at most $\mathcal{O}(nW/\varepsilon)$ different values. Since each vector in $D$ has at most $\mathcal{O}(1/\varepsilon^5\delta^3)$ entries, the running time of the dynamic program is bounded by $(W)^{O(1/\varepsilon^2\delta^3)} \cdot (\delta^2n)^{O(1/\varepsilon^2)} \cdot \mathcal{O}(nW/\varepsilon)^2 \leq (Wn)^{1/\varepsilon^2\mathcal{O}(1/\varepsilon^{16})}$.

After the processing of the dynamic program, we know which items are horizontal and small for a given feasible vector. Therefore, we can place them with the algorithm described in Lemma 11 and NFDH if possible. If it is not possible, we discard the vector and try the next. The overall running time of the algorithm is bounded by $(Wn)^{1/\varepsilon^2\mathcal{O}(1/\varepsilon^{16})}$.

### 5.3 Scheduling contiguous moldable tasks

Again we start with a binary search framework. We use the results from Ludwig and Tiwari [22] to estimate the makespan of the optimal schedule. Afterward, we use the binary search framework to find the makespan of an optimal solution. The guessing steps work the same as in section 5.2 including the guessing of rounded width of horizontal jobs. In the following, we describe how to adjust the dynamic program for this scheduling version.

Let $\psi_j(p) \in M_j$ be the minimal number of processors needed for job $j \in \mathcal{J}$ to have a processing time of at most $p$. We iterate over the non-placed jobs in arbitrary order. Let $j \in \mathcal{J}$. To generate the set $D_j$ in the dynamic program we do the following steps:

- First, we determine for each of the at most $1/\varepsilon^2$ large processing times $p > 1/\mu\text{OPT}$ the number of needed processors $\psi_j(p)$. If this number is smaller than $\delta W$, it is feasible to schedule this job as tall job and we try each possible box for this job and processing time in the dynamic program. Otherwise the job can not have this processing time for the current choice of large and medium sized items.

- Afterward, we determine for each of the at most $1/\varepsilon\delta$ large processing times $p$ with $1/\mu\text{OPT} \geq p > \delta\text{OPT}$ the number of needed processors $\psi_j(p)$. If this number is smaller than $\mu W$, it is feasible to schedule this job as vertical job and we try each possible box for this job and processing time in the dynamic program. Otherwise the job can not have this processing time for the current choice of large and medium sized items.

- Next, we consider the guessed number of processors for horizontal jobs. For two consecutive rounded numbers of machines $m_i, m_{i+1}$, we determine for which number of machines in $M_j \cap \{m_i, \ldots, m_{i+1}\}$ the job has the smallest processing time $\mu_j(m_i)$. If it is smaller than $\mu\text{OPT}$ the job qualifies to be scheduled as a horizontal job. Hence, we make a copy of each vector in $D_{j-1}$ and add the value $\lceil np_j(m_i)/\varepsilon\text{OPT} \rceil$ to the value $h_{m_{i+1}}$.

- After that, we try to schedule the job as a small job. We determine the smallest processing time if the job uses less than $\mu W$ machines. If this processing time is smaller than $\mu(1 + \varepsilon)^2T$ the job can be scheduled as a small job and we try this possibility too, by adding its work to $a_s$ (normalized by $\varepsilon\text{OPT}/n$) to each vector from the set $D_{j-1}$.

- Last, we test if the job can be scheduled as a medium job. We determine the smallest processing time if the job uses between $\mu W$ and $\delta W$ processors. If this processing time is smaller than $\varepsilon(1 + \varepsilon)^2T$
the job can be scheduled as medium job and we add its work (normalized by $\varepsilon\text{OPT}/n$) to a copy of each vector from the set $D_{j-1}$. On the other hand, the job can be scheduled as medium job, if its processing time is between $\mu(1+\varepsilon)^2T$ and $\delta(1+\varepsilon)^2T$. We determine the minimal number of processors, such that the job has a processing time between $\mu(1+\varepsilon)^2T$ and $\delta(1+\varepsilon)^2T$ and we add its work (normalized by $\varepsilon\text{OPT}/n$) to a copy of each vector from the set $D_{j-1}$.

Each vector in the dynamic program has at most $O(1/\varepsilon^5\delta^3)$ entries. The values of the entries are bounded by $W$, $\delta^2n$ and $nW/\varepsilon$. For each job we try at most $O(1/\varepsilon^5\delta^3)$ entries. Therefore, the running time of the dynamic program and the entire algorithm can be bounded by $(Wn)^{1/\varepsilon^2}O(1/\varepsilon^{16})$.

6 Conclusion

In this paper, we have nearly closed the gap between the lower bound of the approximation ratio and best approximation ratio for the problems pseudo-polynomial Strip Packing with and without rotations and the Contiguous Moldable Task Scheduling.

Still open remains the question whether we actually can find algorithms with approximation ratio exactly $5/4$. Furthermore, it would be interesting to study pseudo polynomial algorithms for other two dimensional packing problems as 2D Knapsack or 2D Bin Packing. Concerning polynomial algorithms, there is still a large gap between the lower bound for an absolute approximation ratio of $3/2$ unless $P = NP$ and $5/3+\varepsilon$ which is the best absolute approximation ratio achieved so far [10]. Furthermore, an interesting question for further research is, if we can find better approximations for the case of monotonic moldable jobs. While the lower bound of $5/4$ holds for the general case of scheduling contiguous moldable jobs in pseudo-polynomial time, a PTAS could be possible if we consider monotonic jobs.

References

[1] Anna Adamaszek, Tomasz Kociumaka, Marcin Pilipczuk, and Michal Pilipczuk. Hardness of approximation for strip packing. *ACM Trans. Comput. Theory*, 9(3):14:1–14:7, September 2017. doi:10.1145/3092026.

[2] 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. doi:10.1016/0196-6774(81)90034-1.

[3] 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. doi:10.1137/0209064.

[4] Iwo Bladek, Maciej Drozdowski, Frédéric Guinand, and Xavier Schepler. On contiguous and non-contiguous parallel task scheduling. *J. Scheduling*, 18(5):487–495, 2015. URL: https://doi.org/10.1007/s10951-015-0427-z, doi:10.1007/s10951-015-0427-z.

[5] Marin Bougeret, Pierre-François Dutot, Klaus Jansen, Christina Robenek, and Denis Trystram. Approximation algorithms for multiple strip packing and scheduling parallel jobs in platforms. *Discrete Mathematics, Algorithms and Applications*, 3(4):553–586, 2011. doi:10.1142/S1793830911001413.

[6] Edward G. Coffman Jr., Michael R. Garey, David S. Johnson, and Robert Endre Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9(4):808–826, 1980. doi:10.1137/0209062.

[7] Leah Epstein and Rob van Stee. This side up! *ACM Trans. Algorithms*, 2(2):228–243, 2006. URL: http://doi.acm.org/10.1145/1150334.1150339, doi:10.1145/1150334.1150339.
[8] Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, and Arindam Khan. Improved pseudo-polynomial-time approximation for strip packing. In 36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), pages 9:1–9:14, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.9.

[9] Igal Golan. Performance bounds for orthogonal oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 10(3):571–582, 1981. doi:10.1137/0210042.

[10] Rolf Harren, Klaus Jansen, Lars Prädel, and Rob van Stee. A (5/3 + ε)-approximation for strip packing. *Computational Geometry*, 47(2):248–267, 2014. doi:10.1016/j.comgeo.2013.08.008.

[11] Rolf Harren and Rob van Stee. Improved absolute approximation ratios for two-dimensional packing problems. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques., volume 5687 of Lecture Notes in Computer Science, pages 177–189. Springer, 2009. doi:10.1007/978-3-642-03685-9_14.

[12] Sören Henning, Klaus Jansen, Malin Rau, and Lars Schmarje. Complexity and inapproximability results for parallel task scheduling and strip packing. *CoRR*, abs/1705.04587, 2017. URL: http://arxiv.org/abs/1705.04587.

[13] Klaus Jansen. A (3/2 + ε) approximation algorithm for scheduling moldable and non-moldable parallel tasks. In 24th ACM Symposium on Parallelism in Algorithms and Architectures, (SPAA), pages 224–235, 2012. doi:10.1145/2312005.2312048.

[14] Klaus Jansen and Felix Land. Scheduling monotone moldable jobs in linear time. *CoRR*, abs/1711.00103, 2017. URL: https://arxiv.org/abs/1711.00103.

[15] Klaus Jansen and Malin Rau. Improved approximation for two dimensional strip packing with polynomial bounded width. *CoRR*, abs/1610.04430, 2016. URL: https://arxiv.org/abs/1610.04430v2, arXiv:1610.04430v2.

[16] Klaus Jansen and Malin Rau. Improved approximation for two dimensional strip packing with polynomial bounded width. In WALCOM: Algorithms and Computation, volume 10167 of LNCS, pages 409–420, 2017. doi:10.1007/978-3-319-53925-6_32.

[17] Klaus Jansen and Roberto Solis-Oba. Rectangle packing with one-dimensional resource augmentation. *Discrete Optimization*, 6(3):310–323, 2009. doi:10.1016/j.disopt.2009.04.001.

[18] Klaus Jansen and Ralf Thöle. Approximation algorithms for scheduling parallel jobs. *SIAM Journal on Computing*, 39(8):3571–3615, 2010. doi:10.1137/080736491.

[19] Klaus Jansen and Rob van Stee. On strip packing with rotations. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 755–761. ACM, 2005. doi:10.1145/1060590.1060702.

[20] Mohammad M. Karbasioun, Gennady Shaikhet, Evangelos Kranakis, and Ioannis Lambadaris. Power strip packing of malleable demands in smart grid. In *Proceedings of IEEE International Conference on Communications, ICC 2013, Budapest, Hungary, June 9-13, 2013*, pages 4261–4265. IEEE, 2013. doi:10.1109/ICC.2013.6655233.

[21] Claire Kenyon and Eric Rémiła. A near-optimal solution to a two-dimensional cutting stock problem. *Mathematics of Operations Research*, 25(4):645–656, 2000. doi:10.1287/moor.25.4.645.12118.
[22] Walter Ludwig and Prasoon Tiwari. Scheduling malleable and nonmalleable parallel tasks. In Daniel Dominic Sleator, editor, *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. 23-25 January 1994, Arlington, Virginia., pages 167–176. ACM/SIAM, 1994. URL: http://dl.acm.org/citation.cfm?id=314464.314491.

[23] Flávio Keidi Miyazawa and Yoshiko Wakabayashi. Packing problems with orthogonal rotations. In Martin Farach-Colton, editor, *LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, Buenos Aires, Argentina, April 5-8, 2004, Proceedings*, volume 2976 of *Lecture Notes in Computer Science*, pages 359–368. Springer, 2004. URL: https://doi.org/10.1007/978-3-540-24698-5_40, doi:10.1007/978-3-540-24698-5_40.

[24] Gregory Mounie, Christophe Rapine, and Denis Trystram. A 3/2-approximation algorithm for scheduling independent monotonic malleable tasks. *SIAM J. Comput.*, 37(2):401–412, 2007. doi:10.1137/S0097539701385995.

[25] Giorgi Nadiradze and Andreas Wiese. On approximating strip packing with a better ratio than 3/2. In *27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1491–1510, 2016. doi:10.1137/1.9781611974331.ch102.

[26] Anshu Ranjan, Pramod P. Khargonekar, and Sartaj Sahni. Offline first fit scheduling in smart grids. In *2015 IEEE Symposium on Computers and Communication, ISCC 2015, Larnaca, Cyprus, July 6-9, 2015*, pages 758–763. IEEE Computer Society, 2015. doi:10.1109/ISCC.2015.7405605.

[27] Ingo Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *2nd Annual European Symposium on Algorithms (ESA) - Algorithms*, pages 290–299, 1994. doi:10.1007/BFb0049416.

[28] Daniel Dominic Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Information Processing Letters*, 10(1):37–40, 1980. doi:10.1016/0020-0190(80)90121-0.

[29] A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing*, 26(2):401–409, 1997. doi:10.1137/S0097539793255801.

[30] Maxim Sviridenko. A note on the kenyon-remila strip-packing algorithm. *Information Processing Letters*, 112(1-2):10–12, 2012. doi:10.1016/j.ipl.2011.10.003.

[31] Shaojie Tang, Qiujuan Huang, Xiang-Yang Li, and Dapeng Wu. Smoothing the energy consumption: Peak demand reduction in smart grid. In *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013*, pages 1133–1141. IEEE, 2013. doi:10.1109/INFCOM.2013.6566904.

[32] John Turek, Joel L. Wolf, and Philip S. Yu. Approximate algorithms scheduling parallelizable tasks. In *4th annual ACM symposium on Parallel algorithms and architectures (SPAA)*, pages 323–332, 1992. doi:10.1145/140901.141909.