

On Guillotine Separable Packings for the Two-Dimensional Geometric Knapsack Problem

Arindam Khan ✉ 

Indian Institute of Science, Bangalore, India

Arnab Maiti ✉ 

Indian Institute of Technology, Kharagpur, India

Amatya Sharma ✉ 

Indian Institute of Technology, Kharagpur, India

Andreas Wiese ✉ 

Universidad de Chile, Santiago, Chile

Abstract

In two-dimensional geometric knapsack problem, we are given a set of n axis-aligned rectangular items and an axis-aligned square-shaped knapsack. Each item has integral width, integral height and an associated integral profit. The goal is to find a (non-overlapping axis-aligned) packing of a maximum profit subset of rectangles into the knapsack. A well-studied and frequently used constraint in practice is to allow only packings that are guillotine separable, i.e., every rectangle in the packing can be obtained by recursively applying a sequence of edge-to-edge axis-parallel cuts that do not intersect any item of the solution. In this paper we study approximation algorithms for the geometric knapsack problem under guillotine cut constraints. We present polynomial time $(1 + \varepsilon)$ -approximation algorithms for the cases with and without allowing rotations by 90 degrees, assuming that all input numeric data are polynomially bounded in n . In comparison, the best-known approximation factor for this setting is $3 + \varepsilon$ [Jansen-Zhang, SODA 2004], even in the cardinality case where all items have the same profit.

Our main technical contribution is a structural lemma which shows that any guillotine packing can be converted into another *structured* guillotine packing with almost the same profit. In this packing, each item is completely contained in one of a constant number of boxes and **L**-shaped regions, inside which the items are placed by a simple greedy routine. In particular, we provide a clean sufficient condition when such a packing obeys the guillotine cut constraints which might be useful for other settings where these constraints are imposed.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Approximation Algorithms, Multidimensional Knapsack, Guillotine Cuts, Geometric Packing, Rectangle Packing

Digital Object Identifier 10.4230/LIPIcs.SoCG.2021.48

Related Version *Full Version:* <https://arxiv.org/abs/2103.09735>

Acknowledgements The authors would like to thank Madhusudhan Reddy for helpful discussions. A part of this work was done when Arnab Maiti and Amatya Sharma were Narendra undergraduate summer interns at Indian Institute of Science.

1 Introduction

Geometric packing problems have many important applications in cutting stock [27], VLSI design [32], logistics [13], smart-grids [25], etc. Two-dimensional geometric knapsack (2GK), a multidimensional generalization of the classical knapsack problem, is one of the central problems in this area. We are given a set of n axis-aligned (open) rectangles (also called *items*) $I := \{1, 2, \dots, n\}$, where rectangle i has integral width $w(i)$, integral height $h(i)$



© Arindam Khan, Arnab Maiti, Amatya Sharma, and Andreas Wiese; licensed under Creative Commons License CC-BY 4.0

37th International Symposium on Computational Geometry (SoCG 2021).

Editors: Kevin Buchin and Éric Colin de Verdière; Article No. 48; pp. 48:1–48:17

Leibniz International Proceedings in Informatics



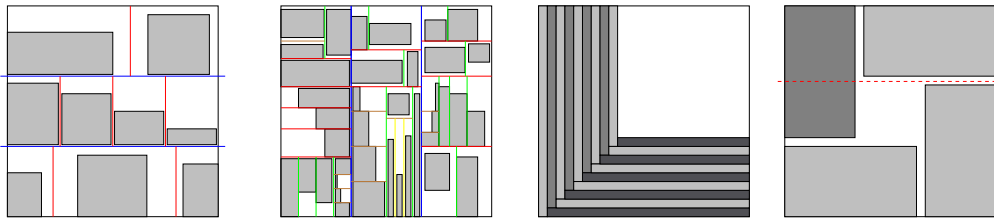
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



and an associated integral profit $p(i)$. We are also given an axis-aligned square knapsack $K := [0, N] \times [0, N]$, where $N \in \mathbb{N}$. The goal is to select a subset of items $I' \subseteq I$ of maximum total profit $p(I') := \sum_{i \in I'} p(i)$ so that they can be packed in the knapsack. The packing needs to be *axis-parallel* and *non-overlapping*, i.e., such packing maps each rectangle $i \in I'$ to a new translated open rectangle $R(i) := (\text{left}(i), \text{right}(i)) \times (\text{bottom}(i), \text{top}(i))$ where $\text{right}(i) = \text{left}(i) + w(i)$, $\text{top}(i) = \text{bottom}(i) + h(i)$, $\text{left}(i) \geq 0$, $\text{bottom}(i) \geq 0$, $\text{right}(i) \leq N$, $\text{top}(i) \leq N$ and for any $i, j \in I'$, we must have $R(i) \cap R(j) = \emptyset$. In 2GK, items are not allowed to be rotated. There is another variant with rotations that we denote by 2GK(R), where items are allowed to be rotated by 90 degrees.

2GK has rich connections with many important problems, such as maximum independent set of rectangles (MISR) [2], 2-D bin packing [7], strip packing [23, 30], mixed packing [39], fair allocation [45], storage allocation [42], unsplittable flow [28], etc. Leung et al. [40] showed that the problem is strongly NP-hard. Jansen and Zhang [35] gave $(2 + \varepsilon)$ -approximation algorithms for both 2GK and 2GK(R), where $\varepsilon > 0$ is an arbitrarily small constant. Finally, Gálvez et al. [24] broke the barrier of 2 by giving a 1.89-approximation algorithm for 2GK and $(3/2 + \varepsilon)$ -approximation algorithm for 2GK(R). Furthermore, if the input data is quasi-polynomially bounded (i.e., $N \leq n^{(\log n)^c}$ for some $c > 0$) then there exists a quasi-polynomial time approximation scheme (QPTAS) for both problems [3]. Polynomial time approximation schemes (PTASs) are known for many special cases: if all items are small [20], if all items are squares [31, 34], if the profit of each item equals its area [5], and if we allow resource augmentation (i.e., the size of the knapsack can be slightly increased) [21, 33]. However, it is an open problem to construct a PTAS, even with pseudo-polynomial running time.

One can view geometric packing as a cutting problem where we are given a large sheet or stock unit (maybe metal, glass, wood, rubber, or cloth), which should be cut into pieces out of the given input set. Cutting technology often only allows axis-parallel end-to-end cuts called *guillotine cuts*. See [8, 49] for practical applications and software related to guillotine packing. In this setting, we seek for solutions in which we can cut out the individual objects by a recursive sequence of guillotine cuts that do not intersect any item of the solution. The related notion of *k-stage packing* was originally introduced by Gilmore and Gomory [27]. Here each stage consists of either vertical or horizontal guillotine cuts (but not both). On each stage, each of the sub-regions obtained on the previous stage is considered separately and can be cut again by using horizontal or vertical guillotine cuts. In *k-stage packing*, the number of cuts to obtain each rectangle from the initial packing is at most k , plus an additional cut to trim (i.e., separate the rectangles itself from a waste area). Intuitively, this means that in the cutting process we change the orientation of the cuts $k - 1$ times. The case where $k = 2$, usually referred to as *shelf packing*, has been studied extensively.



■ **Figure 1** The first three packing are guillotine separable packings of 2-stages, 5-stages, and many stages, respectively. The last packing is not a guillotine packing as any end-to-end cut in the knapsack intersects at least one of the packed rectangles.

In this paper, we study the two-dimensional knapsack problem under guillotine cuts (2GGK). The input is the same as for 2GK, but we require additionally that the items in the solution can be separated by a sequence of guillotine cuts, and we say that then they are *guillotine separable*. NP-hardness of 2GGK follows from a reduction from the (one-dimensional) knapsack problem. Christofides et al. [14] studied the problem in 1970s. Since then many heuristics have been developed to efficiently solve benchmark instances, based on tree-search [50], branch-and-bound [29], dynamic optimization [9], tabu search [4], genetic algorithms [44], etc. Despite a staggering number of recent experimental papers [10, 15, 16, 18, 19, 22, 41, 51], there was little theoretical progress for 2GGK, due to limitations of past techniques. Since 2004, the $(3 + \varepsilon)$ -approximation for 2GK by Jansen and Zhang [35] has been the best-known approximation algorithm for 2GGK. Recently, Abed et al. [1] have studied approximation algorithms for the cardinality cases of 2GGK and 2GGK(R) and have given a QPTASs, assuming the input data to be quasi-polynomially bounded.

Most algorithms for 2GK utilize a *container packing* (see Section 2) which arranges the items in the knapsack such that they are packed inside a constant number of axis-aligned boxes (containers). The best sizes and locations of these containers can be guessed efficiently since there are only a constant number of them. Then inside each container the items are packed either in one-stage packings or in two-stage packings (if items are small). However, Gálvez et al. [24] show that one cannot obtain a better approximation ratio than 2 with container-based packings with only $O(1)$ many containers, due to interaction between *horizontal* (wide and thin) and *vertical* (tall and narrow) items. To break this barrier, they use a *corridor-decomposition* where the knapsack is divided into axis-parallel polygonal regions called *corridors* with constant number of regions called *subcorridors*. Vertical (resp. horizontal) items are packed in only vertical (resp. horizontal) subcorridors. After simplifying the interaction between vertical and horizontal items, they define two types of packings. In one packing, they *process* the subcorridors to obtain a container-based packing. In the other, a profitable subset of long horizontal and long vertical items are packed in an **L**-shaped region. They prove that the best of these two packings achieves a better approximation ratio than 2. However, it is not clear how to use this approach for 2GGK: even if we start with an optimal guillotine packing, the rearrangements of items may not preserve guillotine separability, and hence they might not lead to a feasible solution to 2GGK.

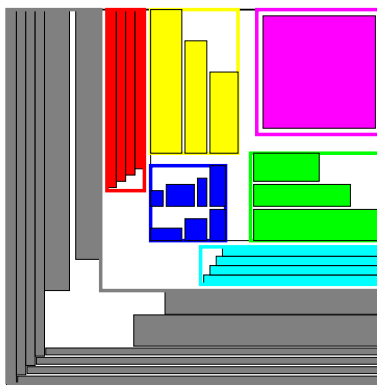
1.1 Our contribution

In this paper, we obtain $(1 + \varepsilon)$ -approximation algorithms with pseudo-polynomial running time for both 2GGK and 2GGK(R), i.e., the running time is a polynomial if the (integral) input numbers are all polynomially bounded in n . The key idea is to show that there are $(1 + \varepsilon)$ -approximate solutions in which the knapsack is divided into simple *compartments* that each have the shape of a rectangular box or an **L**, see Figure 2. Inside each compartment, the items are placed in a very simple way, e.g., all horizontal items are simply stacked on top of each other, all vertical items are placed side by side, and all small items are packed greedily with the Next-Fit-Decreasing-Height algorithm [17], see Figure 2. To establish this structure, we crucially exploit that the optimal solution is guillotine separable; in particular, in 2GK (where the optimal solution might not be guillotine separable) more complicated compartments may be necessary for near-optimal solutions, e.g., with the form of a ring.

While the items in our structured solution are guillotine separable, we cannot separate the compartments by guillotine cuts since we cannot cut out an **L**-shaped compartment with such cuts. This makes it difficult to compute a solution of this type since it is not sufficient to ensure that (locally) within each compartment the items are guillotine separable (which is

immediately guaranteed by our simple packings inside them). Therefore, our compartments have an important additional property: they can be separated by a *pseudo guillotine cutting sequence*. This is a cutting sequence in which each step is either a guillotine cut, or a cut along two line segments that separates a rectangular area into an **L**-shaped compartment and a smaller rectangular area, see Figure 5. We prove a strong property for compartments that admit such a pseudo guillotine cutting sequence: we show that if we pack items into such compartments in the simple way mentioned above, this will *always* yield a solution that is *globally* guillotine separable. This property and our structural result might have applications also in other settings where we are interested in solutions that are guillotine separable.

Our strong structural result allows us to construct algorithms (for the cases with and without rotations) that are relatively simple: we first guess the constantly many compartments in the structured solution mentioned above. Then we compute up to a factor $1 + \varepsilon$, the most profitable set of items that can be placed nicely into them, using a simplified version of a recent algorithm in [26]. The resulting solutions use up to $\Theta(\log(nN))$ stages (unlike e.g., solutions of the Next-Fit-Decreasing-Height algorithm [17] that need only two stages). We prove a lower bound, showing that there is a family of instances of 2GGK that does not admit $(2 - \varepsilon)$ -approximate solutions with only $o(\log N)$ -stages.



■ **Figure 2** A structured packing of items into compartments that each have the shape of an **L**- or a rectangular box.

1.2 Other related work

There are many well-studied geometric packing problems. In the 2D bin packing problem (2BP), we are given a set of rectangular items and unit square bins, and the goal is to pack all the items into a minimum number of bins. The problem is APX-hard [6] and the currently best known approximation ratio is 1.405 [7]. In the 2D strip packing problem (2SP), we are given a set of rectangular items and a fixed-width unbounded-height strip, and the goal is to pack all the items into the strip such that the height of the strip is minimized. Kenyon and Rémila gave an APTAS for the problem [36] using a 3-stage packing.

Both 2BP and 2SP are well-studied in the guillotine setting [46]. Caprara [11] gave a 2-stage T_∞ (≈ 1.691)-approximation for 2BP. Afterwards, Caprara et al. [12] gave an APTAS for 2-stage 2BP and 2-stage 2SP. Later, Bansal et al. [8] showed an APTAS for guillotine 2BP. Bansal et al. [7] conjectured that the worst-case ratio between the best guillotine 2BP and the best general 2BP is $4/3$. If true, this would imply a $(\frac{4}{3} + \varepsilon)$ -approximation algorithm for 2BP. Seiden et al. [47] gave an APTAS for guillotine 2SP. Both the APTAS for guillotine

2BP and guillotine 2SP are based on the fact that general guillotine 2BP or guillotine 2SP can be approximated arbitrarily well by $O(1)$ -stage packings, and such $O(1)$ -stage packings can be found efficiently. Interestingly, we showed that this property is not true for 2GGK.

Pach and Tardos [43] conjectured that, for any set of n non-overlapping axis-parallel rectangles, there is a guillotine cutting sequence separating $\Omega(n)$ of them. Recently, the problem has received attention in [1,38] since, if true, this would imply a $O(1)$ -approximation for the Maximum Independent Set of Rectangles problem, a long-standing open problem.

2 Methodology

For simplicity of presentation, we primarily focus on the cardinality case, i.e., assume that $p(i) = 1$ for each item $i \in I$. For a detailed description of the generalization to arbitrary item profits, see [37]. For each $n \in \mathbb{N}$ we define $[n] := \{1, 2, \dots, n\}$.

We classify the input items according to their heights and widths. For two constants $1 \geq \varepsilon_{large} > \varepsilon_{small} > 0$ to be defined later, we classify each item $i \in I$ as:

- *Large*: $w_i > \varepsilon_{large}N$ and $h_i > \varepsilon_{large}N$;
- *Small*: $w_i \leq \varepsilon_{small}N$ and $h_i \leq \varepsilon_{small}N$;
- *Horizontal*: $w_i > \varepsilon_{large}N$ and $h_i \leq \varepsilon_{small}N$;
- *Vertical*: $h_i > \varepsilon_{large}N$ and $w_i \leq \varepsilon_{small}N$;
- *Intermediate*: Either $\varepsilon_{large}N \geq h_i > \varepsilon_{small}N$ or $\varepsilon_{large}N \geq w_i > \varepsilon_{small}N$.

Using standard shifting arguments, one can show that we can ignore intermediate items.

► **Lemma 1** ([24]). *Let $\varepsilon > 0$ and $f(\cdot)$ be any positive increasing function such that $f(x) < x \forall x \in (0, 1]$. Then we can efficiently find $\varepsilon_{large}, \varepsilon_{small} \in \Omega_\varepsilon(1)$, with $\varepsilon \geq f(\varepsilon) \geq \varepsilon_{large} \geq f(\varepsilon_{large}) \geq \varepsilon_{small}$ so that the total profit of intermediate rectangles is at most $\varepsilon p(OPT)$.*

We define *skewed* items to be items that are horizontal or vertical. Let $I_{large}, I_{small}, I_{hor}, I_{ver}, I_{skew}$ be the set of large, small, horizontal, vertical, and skewed rectangles, respectively. The corresponding intersections with OPT (the optimal guillotine packing) defines the sets $OPT_{large}, OPT_{small}, OPT_{hor}, OPT_{ver}, OPT_{skew}$, respectively.

2.1 Compartments

Our goal is to partition the knapsack into compartments, such that there is an $(1 + \varepsilon)$ -approximate solution whose items are placed in a structured way inside these compartments. We will use two types of compartments: *box-compartments* and **L**-compartments.

► **Definition 1** (Box-compartment). *A box-compartment B is an axis-aligned rectangle that satisfies $B \subseteq K := [0, N] \times [0, N]$.*

► **Definition 2** (L-compartment). *An L-compartment L is a subregion of K bounded by a simple rectilinear polygon with six edges e_0, e_1, \dots, e_5 such that for each pair of horizontal (resp. vertical) edges e_i, e_{6-i} with $i \in \{1, 2\}$ there exists a vertical (resp. horizontal) line segment ℓ_i of length less than $\varepsilon_{large}N/2$ such that both e_i and e_{6-i} intersect ℓ_i but no other edges intersect ℓ_i .*

Since the length of the line segments ℓ_i is less than $\varepsilon_{large}N/2$, this implies that inside an L-compartment L we cannot place large items, inside the horizontal part of L we cannot place vertical items, and inside the vertical part of L we cannot place horizontal items.

We seek for a structured packing inside of these compartments according to the following definitions. Inside box-compartments, we want only one type of items and we want that the skewed items are placed in a very simple way, see Figure 2.

► **Definition 3.** Let B be a box-compartment and let $I_B \subseteq I$ be a set of items that are placed non-overlappingly inside B . We say that the placement of I_B is nice if the items in I_B are guillotine separable and additionally

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

Inside **L**-compartments we allow only skewed items and we want them to be placed in a similar way as in the boxes, see Figure 2 and 3.

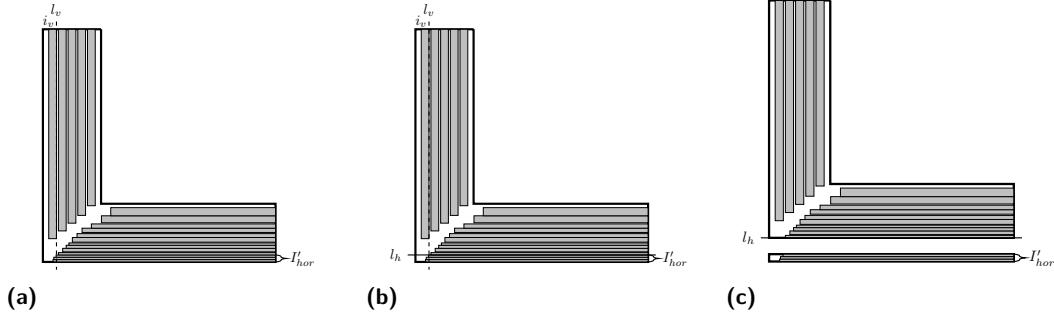
► **Definition 4.** Let L be an **L**-compartment and let $I_L \subseteq I$ be a set of items that are placed non-overlappingly inside L . We say that the placement of I_L is nice if

- $I_L \subseteq I_{skew}$, and
- the items in $I_L \cap I_{hor}$ are stacked on top of each other inside L , and
- the items in $I_L \cap I_{ver}$ are stacked side by side inside L .

A nice placement inside an **L**-compartment yields a guillotine separable packing.

► **Lemma 2.** Consider a set of items $I_L \subseteq I$ that is placed nicely inside an **L**-compartment L . Then I_L is guillotine separable.

Proof sketch. One can show that there always exists a guillotine cut that separates one or more horizontal or vertical items in I_L from the other items in I_L , see Figure 3. Then this argument is applied recursively. ◀



■ **Figure 3** (a) A nicely packed set of skewed items inside an **L**-compartment. The vertical cut l_v separates the leftmost vertical item i_v from the other vertical items but it intersects the horizontal items in I'_{hor} . (b) However, then the horizontal cut l_h separates the items in I'_{hor} from the other horizontal items without intersecting any vertical item. (c) The corresponding guillotine cut that partitions the **L**-compartment into a box-compartment and a smaller **L**-compartment.

2.2 Pseudo-guillotine separable compartments

We seek to partition the knapsack into box- and **L**-compartments and then place items into these compartments. We also want to ensure that the resulting solution is guillotine separable. We could guarantee this if there was a guillotine cutting sequence that separates all compartments and require that the items inside the compartments are placed nicely. Then, we could first separate all compartments by the mentioned cutting sequence and then separate the items inside of each compartment by guillotine cuts (as they are packed nicely).

However, there is no guillotine cutting sequence that cuts out an **L**-compartment from the knapsack since no guillotine cut can separate the **L**-compartment from the area at the “inner” part of the **L**-compartment. Therefore, we require for the compartments in our knapsack only that there is a *pseudo-guillotine cutting sequence*. A pseudo-guillotine cutting sequence has the following two operations (see Figure 5): given a rectangle $R \subseteq K$ it

- applies a horizontal or vertical guillotine cut that separates R into two disjoint rectangles R_1, R_2 and then continues recursively with R_1 and R_2 , or
- for an **L**-compartment $L \subseteq R$ such that $R \setminus L$ is a rectangle, it partitions R into L and $R \setminus L$ and then continues recursively with $R \setminus L$ (but not with L). Note that we cannot do this operation with every **L**-compartment $L' \subseteq R$ since possibly $R \setminus L'$ is not a rectangle.

We formalize this in the following definition.

► **Definition 5.** A pseudo-guillotine cutting sequence (for compartments) for a set of compartments \mathcal{C} is a binary tree $T = (V, E)$ where for each vertex $v \in V$ there is an associated shape $S_v \subseteq K$ such that

- for the root $r \in V$ of T it holds that $S_r = K$,
- for each internal vertex v with children u, w it holds that
 - S_v is a rectangle with $S_v = S_u \dot{\cup} S_w$ (so in particular S_u and S_w are disjoint),
 - either S_u and S_w are both rectangles or one of them is an **L**-compartment and the other is a rectangle,
- for each compartment $C \in \mathcal{C}$ there is a leaf $v \in V$ such that $S_v = C$.

Observe that each **L**-compartment corresponds to a leaf node in T .

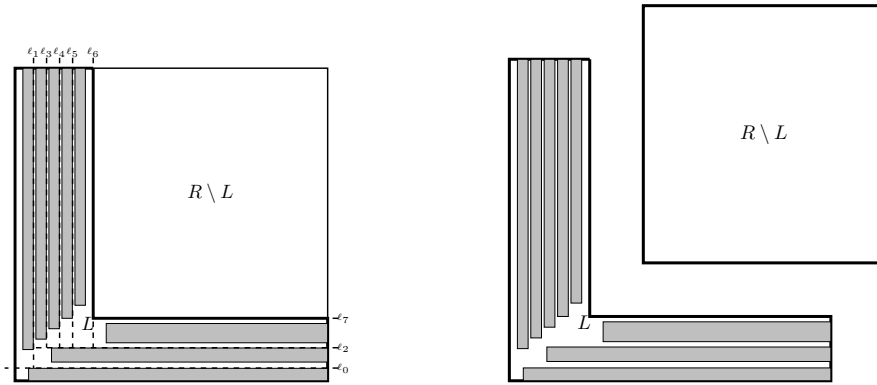
Now the important insight is that if a set of compartments \mathcal{C} admits a pseudo-guillotine cutting sequence, then *any nice* placement of items inside these compartments is guillotine separable (globally). In particular, given such compartments \mathcal{C} , we can place items inside the compartments in \mathcal{C} without needing to worry whether the resulting packing will be guillotine separable globally, as long as we place these items nicely. Intuitively, this is true since we can use the cuts of the pseudo-guillotine cutting sequence as a template for a global cutting sequence for the items: whenever the former sequence

- makes a guillotine cut, we simply do the same cut,
- when it separates an **L**-compartment L from a rectangular region R , we separate the items inside L by a sequence of guillotine cuts; it turns out that we can do this since all items inside L are placed nicely and skewed.

Finally, we separate the items inside each box-compartment B by guillotine cuts, using the fact that the items inside B are placed nicely.

► **Lemma 3.** Let \mathcal{C} be a set of compartments inside K that admit a pseudo-guillotine cutting sequence. Let $I' \subseteq I$ be a set of items that are placed nicely inside the compartments in \mathcal{C} . Then there is a guillotine cutting sequence for I' .

Proof sketch. Let P denote the pseudo-guillotine cutting sequence. We construct a guillotine cutting sequence for I' based on P . We follow the cuts of P . Whenever P makes a guillotine cut, then we also do this guillotine cut. When P separates an **L**-compartment L from a rectangular region R , then we apply a sequence of guillotine cuts that step by step separates all items in L from $R \setminus L$. Since inside L the items are placed nicely, one can show that there exist such cuts that don't intersect any item in $R \setminus L$ (see Figure 4). ◀



■ **Figure 4** Partition of rectangle R into L and $R \setminus L$ when items inside L are packed nicely. ℓ_0, \dots, ℓ_7 (dashed lines) are a sequence of guillotine cuts that ultimately separate out the items in L from R .

2.3 Near-optimal structured solutions

Our main technical contribution is to show that there exists a $(1 + \varepsilon)$ -approximate solution whose items can be placed nicely inside a set of compartments \mathcal{C} that admit a pseudo-guillotine cutting sequence. By Lemma 3 there is a guillotine cutting sequence for them.

► **Lemma 4.** *There exists a set $OPT' \subseteq I$ and a partition of K into a set of $O_\varepsilon(1)$ ¹ compartments \mathcal{C} such that*

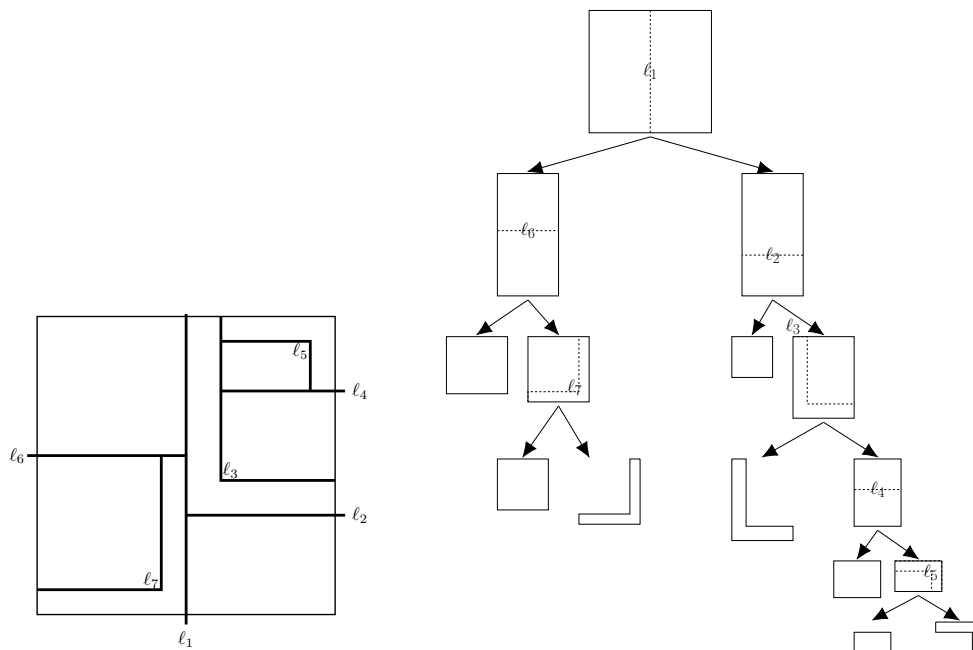
- $|OPT'| \geq (1 - \varepsilon)|OPT|$,
- *the compartments \mathcal{C} admit a pseudo-guillotine cutting sequence,*
- *the items in OPT' can be placed nicely inside the compartments \mathcal{C} .*

We will prove Lemma 4 in Section 3. Our main algorithm works as follows. First, we guess the $O_\varepsilon(1)$ compartments \mathcal{C} due to Lemma 4 in time $(nN)^{O_\varepsilon(1)}$ (note that we can assume w.l.o.g. that they have integral coordinates). Then we place items nicely inside \mathcal{C} while maximizing the cardinality of the placed items. For this we use a $(1 + \varepsilon)$ -approximation algorithm which is a slight adaptation of a recent algorithm in [26] for the 2GK problem (i.e., without requiring that the computed solution is guillotine separable). In fact, we simplify some steps of that algorithm since our compartments are very simple.

► **Lemma 5.** *Given a set of compartments \mathcal{C} . In time $(nN)^{O_\varepsilon(1)}$ we can compute a set of items $ALG \subseteq I$ that are placed nicely inside \mathcal{C} such that $|ALG| \geq (1 - \varepsilon)|OPT'|$ for any set of items OPT' that can be placed nicely inside the compartments \mathcal{C} . Inside each compartment $C \in \mathcal{C}$ the set ALG admits an $O_\varepsilon(\log(nN))$ -stage packing.*

We will prove Lemma 5 in Section 4. Then, Lemmas 3, 4, and 5 imply our main theorem for the cardinality case. Due to Lemma 4, our pseudo-guillotine cutting sequence has $O_\varepsilon(1)$ leaf nodes and each of them is either a box- or an **L**-compartment. The packing algorithm due to Lemma 5 gives a $O_\varepsilon(\log(nN))$ -stage packing inside each compartment. This yields globally a $O_\varepsilon(\log(nN))$ -stage packing. For the analysis of our $(1 + \varepsilon)$ -approximation algorithm for the weighted case, see [37].

¹ The notation $O_\varepsilon(f(n))$ means that the implicit constant hidden by the big O notation can depend on ε .



■ **Figure 5** (a) A pseudo-guillotine cutting sequence. The first cut is l_1 , and then the resulting right piece is further subdivided by l_2, l_3, l_4 and l_5 . Similarly, l_6, l_7 subdivide the left piece. Note that l_3, l_5 and l_7 are not guillotine cuts, but they cut out the corresponding L-compartments. (b) step by step pseudo-guillotine cutting sequence corresponding to Figure (a). Dashed line at each level indicates a partition of a rectangle into two regions (two boxes, or one box and one L-shaped).

► **Theorem 6.** *There is a $(1 + \varepsilon)$ -approximation algorithm for 2GGK with a running time of $(nN)^{O_\varepsilon(1)}$ that computes an $O_\varepsilon(\log(nN))$ -stage packing.*

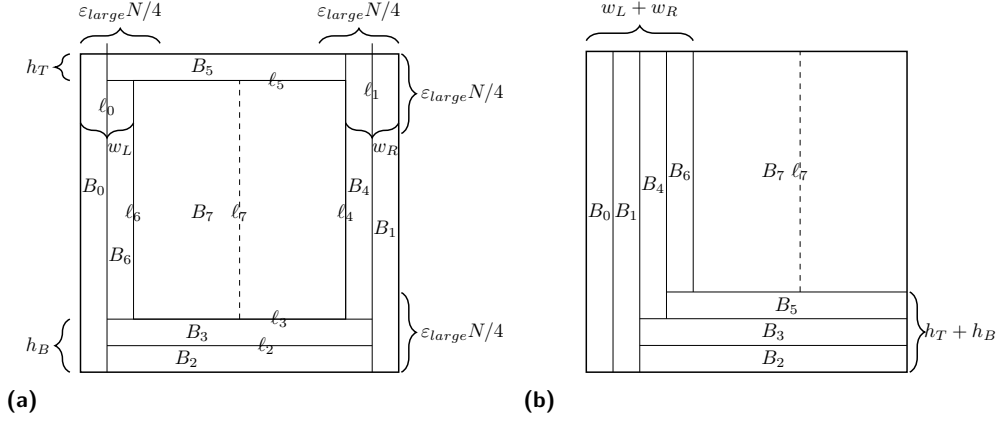
We obtain a similar result also for the rotational case: our structural result from Lemma 4 still holds and the algorithm due to Lemma 5 needs only some minor modifications.

► **Theorem 7.** *There is a $(1 + \varepsilon)$ -approximation algorithm for 2GGK(R) with a running time of $(nN)^{O_\varepsilon(1)}$ that computes an $O_\varepsilon(\log(nN))$ -stage packing.*

3 Existence of near-optimal structured solutions

In this section, we prove Lemma 4 in the cardinality case, i.e., there exists a $(1 + \varepsilon)$ -approximate solution whose items can be placed nicely inside a set of compartments \mathcal{C} that admit a pseudo-guillotine cutting sequence. Note that Lemma 4 trivially holds if $|OPT| \leq O_\varepsilon(1)$ and hence we assume that $|OPT|$ is larger than any given constant (thus we can drop any set of $O_\varepsilon(1)$ items from OPT while losing only a factor of $1 + \varepsilon$).

Consider an optimal solution OPT and a corresponding guillotine cutting sequence S . Temporarily, we remove from the packing the items in OPT_{small} ; we will put back most of them later. We identify a set of cuts of S as follows. Let l_0 denote the first cut of S . Assume w.l.o.g. that l_0 is vertical. If the distance of l_0 to the left and to the right edge of K is at least $\varepsilon_{large}N/4$ then we stop. Otherwise l_0 cuts K into two rectangles R_1, R_2 and assume w.l.o.g. that the width of R_1 is at most $\varepsilon_{large}N/4$. Now we consider how S continues within R_2 . We continue recursively. Assume inductively that we identified a set of cuts l_0, \dots, l_{k-1} of S and suppose that l_{k-1} is vertical cut with distance less than $\varepsilon_{large}N/4$ to the left or the



■ **Figure 6** Transformation to obtain an **L**-compartment.

right edge of K , or that ℓ_{k-1} is horizontal cut with distance less than $\varepsilon_{large}N/4$ to the top or the bottom edge of K . Assume w.l.o.g. that ℓ_{k-1} is vertical with distance less than $\varepsilon_{large}N/4$ to the left edge of K . Then the cut ℓ_{k-1} yields two rectangles R_1, R_2 , and assume that R_1 lies on the left of R_2 . Then we define ℓ_k to be the next cut of S within R_2 . If the distance of ℓ_k to the top *and* the bottom edge of K is at least $\varepsilon_{large}N/4$ then we stop. Otherwise we continue iteratively. Eventually, this procedure must stop, let ℓ_0, \dots, ℓ_k denote the resulting sequence. Let B_0, \dots, B_{k-1} denote the rectangles that are cut off by $\ell_0, \dots, \ell_{k-1}$ and into which we did *not* recurse when we defined ℓ_1, \dots, ℓ_k . Let B_k denote the rectangle that is cut by ℓ_k . Then each rectangle B_i with $i \in \{1, \dots, k-1\}$ satisfies that $w(B_i) \leq \varepsilon_{large}N/4$ or $h(B_i) \leq \varepsilon_{large}N/4$ and in particular cannot contain both horizontal and vertical items. Also, the items of OPT inside B_i are guillotine separable. The important insight is that we can rearrange the rectangles B_0, \dots, B_k (while moving their items accordingly) such that B_0, \dots, B_{k-1} lies in an **L**-compartment $L \subseteq K$ such that $K \setminus L$ is a rectangle, i.e., L lies at the boundary of K as shown in the Figure 6.

► **Lemma 8.** *There exists an **L**-compartment $L \subseteq K$ such that $K \setminus L$ is a rectangle and we can rearrange the rectangles B_0, \dots, B_k such that*

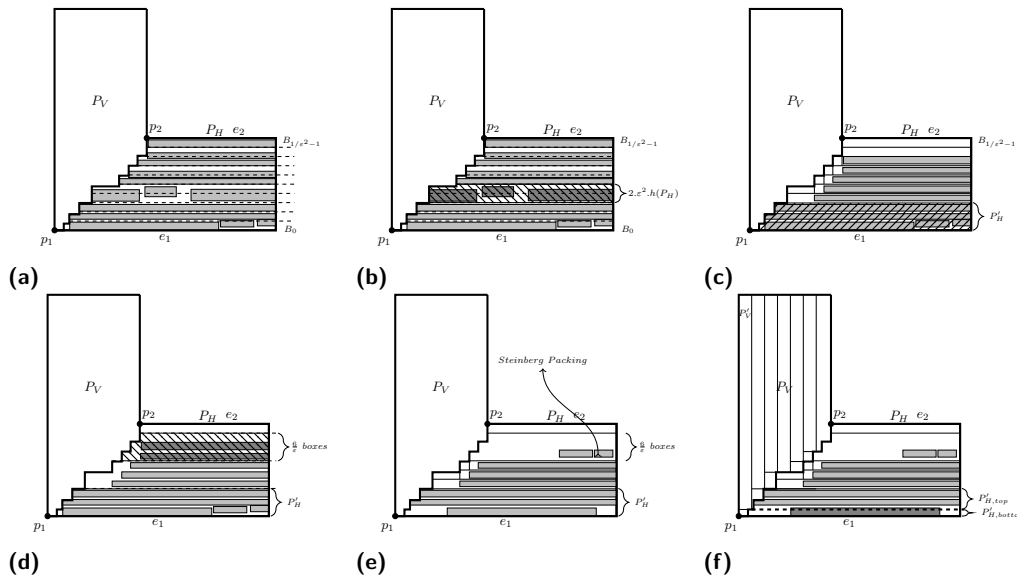
- B_0, \dots, B_{k-1} fit non-overlappingly into L ,
- there is a guillotine cutting sequence for B_0, \dots, B_{k-1} ,
- B_k fits into $K \setminus L$.

Proof. Following the cutting sequence S as described, let us assume that $B_k := [w_L, N - w_R] \times [h_B, N - h_T]$, where $0 \leq w_R, w_L, h_T, h_B \leq \varepsilon_{large}N/4$. Therefore, the cuts $\ell_1, \dots, \ell_{k-1}$ separate of a ring-like region $Q := ([0, w_L] \times [0, N]) \cup ([N - w_R, N] \times [0, N]) \cup ([0, N] \times [0, h_B]) \cup ([0, N] \times [N - h_T, h_T])$ (see Figure 6). Note that some of the values w_R, w_L, h_T, h_B might be 0. The rectangles B_0, \dots, B_{k-1} fit in Q and we want to show that we can rearrange the rectangles in B_0, \dots, B_{k-1} into an **L**-compartment $L \subseteq K$ such that $L := ([0, w_L + w_R] \times [0, N]) \cup ([0, N] \times [0, h_B + h_T])$ and there is a guillotine cutting sequence for B_0, \dots, B_{k-1} . Clearly, B_k fits into $K \setminus L$. We prove the claim by induction on k . The base case is trivial. W.l.o.g. assume the vertical cut ℓ_0 that divides K into B_0, R' , where B_0 lies on the left of R' . Hence, $B_0 := [0, b_0] \times [0, N]$ and $R' := K \setminus B_0$. We use induction on R' to find a packing of B_1, \dots, B_{k-1} in $L' := [b_0, w_L + w_R] \times [0, N] \cup [0, N] \times [0, h_B + h_T]$. Therefore, adding B_0 to L' yields the desired **L**-compartment L . For the guillotine cutting sequence

for B_0, \dots, B_{k-1} , we follow ℓ_0 and afterwards the guillotine cutting sequence for B_1, \dots, B_{k-1} obtained by induction from R' . The other cases, i.e., when B_0 lies right or top or bottom of R' , follow analogously. ◀

We adjust the packing of OPT according to Lemma 8, i.e., for each rectangle B_i with $i \in \{0, \dots, k\}$ we move its items according to where B_i was moved due to the lemma. The resulting packing inside L might not be nice. However, we can fix this by dropping at most $O_\varepsilon(1)$ items and subdividing L into $O_\varepsilon(1)$ box-compartments and a smaller **L**-compartment $L' \subseteq L$ that lies at the outer boundary of L , i.e., such that $L \setminus L'$ is again an **L**-compartment and $h(L') = h(L)$ and $w(L') = w(L)$.

- **Lemma 9.** *Given an **L**-compartment L containing a set of items $I(L)$. There exists a partition of L into one **L**-compartment $L' \subseteq L$ and $O_\varepsilon(1)$ box-compartments $\mathcal{B}(L)$ such that*
- *L' lies at the outer boundary of L ,*
 - *the box-compartments in $\mathcal{B}(L)$ are guillotine separable, and*
 - *there is a nice placement of a set of items $I'(L) \subseteq I(L)$ with $|I'(L)| \geq (1 - \varepsilon)|I(L)| - O_\varepsilon(1)$ inside $\mathcal{B}(L)$ and L' .*



■ **Figure 7** Processing done in Lemma 9 to obtain a nice packing in **L**-compartment.

Proof sketch. Since it is sufficient to place $(1 - \varepsilon)|I'(L)| - O_\varepsilon(1)$ items, we can drop $O_\varepsilon(1)$ items. So w.l.o.g. assume that $I(L)$ contains only skewed items (i.e., we remove all large items). Intuitively, we partition L into two polygons P_H and P_V that are separated via a monotone axis-parallel curve connecting the two vertices of L at the bend of L , such that P_H contains all horizontal items placed inside L and P_V contains all vertical items inside L , see Figure 7a. We rearrange the items in P_H and P_V separately, starting with P_H . Denote by $I(P_H) \subseteq I(L)$ the items of $I(L)$ placed inside P_H .

We place $1/\varepsilon^2$ boxes inside P_H of height $\varepsilon^2 h(P_H)$ each, stacked one on top of the other. We define their width maximally large such that they are still contained inside P_H (note that some area of P_H is then not covered by these boxes), see Figure 7b. Denote by $\{B_0, \dots, B_{1/\varepsilon^2-1}\}$ these boxes in this order, such that B_0 touches the longer horizontal edge of P_H . With a shifting argument, we can show that there are two consecutive boxes B_{j^*}, B_{j^*+1} with

$j^* \leq 1/\varepsilon$ that intersect with at most an $O_\varepsilon(1) + O(\varepsilon|I(P_H)|)$ items in $I(P_H)$. We remove these items. Let $P'_H \subseteq P_H$ denote the part of P_H underneath B_{j^*} (see Figure 7c). We move down by $\varepsilon^2 h(P)$ units each item in $I(P_H)$ that intersect one of the boxes $B_{j^*+2}, \dots, B_{1/\varepsilon^2-1}$ and we remove all $O_\varepsilon(1)$ items from $I(L)$ that intersect more than one box. Note that then the moved items fit into the boxes $\mathcal{B}' := \{B_{j^*+1}, \dots, B_{1/\varepsilon^2-2}\}$.

Using another shifting step, we delete all items in $6/\varepsilon$ consecutive boxes of \mathcal{B}' ; since there are $\Omega(1/\varepsilon^2)$ boxes in \mathcal{B}' this costs only a factor $1 + O(\varepsilon)$ in the profit. We use the empty space to place in it all items in P'_H that are shorter than the shorter horizontal edge of P_H , see Figure 7e. One can show that they can be placed into this empty space using Steinberg's algorithm [48] (maintaining guillotine separability) since the available space is much larger than the area of the items to be placed. For the remaining items in P'_H one can show that the width of each of them is more than half of the width of L . Hence, we can assume w.l.o.g. that they are placed nicely within P'_H . Again, we remove all items that intersect more than one box after this movement, which are at most $O_\varepsilon(1)$ items. Denote by \mathcal{B}_{hor} the resulting set of boxes.

We do a symmetric procedure for P_V , yielding a set of boxes \mathcal{B}_{ver} and a nicely packed region P'_V . Intuitively, we want to define L' as $P'_H \cup P'_V$. However, $P'_H \cup P'_V$ might not have exactly the shape of an **L**-compartment. Nevertheless, one can show that we can subdivide one of these polygons, say P'_H , along a horizontal line into two subpolygons $P'_{H,top}, P'_{H,bottom}$ (with $P'_{H,top}$ lying on the top of $P'_{H,bottom}$) such that

- we can place the items in $P'_{H,top}$ into another set of $O_\varepsilon(1)$ boxes \mathcal{B}'_{hor} that are non-overlapping with $\mathcal{B}_{hor} \cup \mathcal{B}_{ver}$, and
- $L' := P'_{H,bottom} \cup P'_V$ forms an **L**-compartment, see Figure 7f.

Then the items are nicely placed inside L' . To each of the $O_\varepsilon(1)$ boxes $B \in \mathcal{B}_{hor} \cup \mathcal{B}'_{hor} \cup \mathcal{B}_{ver}$ we apply a standard routine that removes some of the items inside B and partitions B into smaller boxes, such that the remaining items inside these smaller boxes are nicely placed. ◀

Therefore, we define that the first cuts of our pseudo-guillotine cutting sequence S' looks as follow: we first separate K into L' and $K \setminus L'$ and then separate the boxes in $\mathcal{B}(L)$. Then we apply a guillotine cut to the rectangular area $K \setminus L$ that corresponds to ℓ_k (since we moved the items in B_k we need to adjust ℓ_k accordingly), which yields two rectangular areas R_1, R_2 . With each of them we continue recursively, i.e., we apply the same routine that we had applied to K above.

We do not recurse further if for a considered rectangular area R it holds that $h(R) < \varepsilon_{large}N$ or $w(R) < \varepsilon_{large}N$. In this case R contains only horizontal or only vertical items, respectively. However, these items might not be packed nicely. Thus, we apply to R a similar routine as in Lemma 9. In a sense, R behaves like a degenerate **L**-compartment with only four edges. Also note that R is a box-compartment.

► **Lemma 10** ([37]). *Given a box-compartment B containing a set of items $I(B)$ with $h(B) < \varepsilon_{large}N$ or $w(B) < \varepsilon_{large}N$, there exists a partition of B into $O_\varepsilon(1)$ box-compartments $\mathcal{B}(B)$ such that*

- *the box-compartments in $\mathcal{B}(B)$ are guillotine separable, and*
- *there is a nice placement of a set of items $I'(B) \subseteq I(B)$ with $|I'(B)| \geq (1-\varepsilon)|I(B)| - O_\varepsilon(1)$ inside $\mathcal{B}(B)$.*

It remains to put back the (small) items in OPT_{small} . Intuitively, we assign them to the empty space in our $O_\varepsilon(1)$ constructed compartments. More formally, we subdivide our compartments further into smaller compartments by guillotine cuts, some of the resulting

compartments are empty, and into those we assign the small items with the Next-Fit-Decreasing-Height algorithm [17]. For each of these compartments we ensure that their height and width are $\varepsilon_{small}N/\varepsilon$. There might be empty space that is not used in this way, however, we can ensure that its total area is very small, e.g., at most $O(\varepsilon^2N^2)$. This allows us to pack essentially all items in OPT_{small} (handling a few special cases differently, e.g., if the total area of the items in OPT_{small} is very small).

Let S' denote the resulting pseudo-guillotine cutting sequence. We need to argue that this yields in total $O_\varepsilon(1)$ compartments. This follows easily since every time we identify a sequence of cuts ℓ_0, \dots, ℓ_k of S , we construct exactly one **L**-compartment and $O_\varepsilon(1)$ box-compartments. Also, after each such operation, we recurse on rectangular areas R_1, R_2 that are at least by $\varepsilon_{large}N/4$ units thinner or shorter (i.e., by at least $\varepsilon_{large}N/4$ units smaller in one of the two dimensions) than the rectangular area that we had started with when we constructed ℓ_0, \dots, ℓ_k (which is the whole knapsack K in the first iteration). Also, when we do not recurse further we subdivide the remaining region into $O_\varepsilon(1)$ box-compartments. Each resulting compartment is subdivided into $O_\varepsilon(1)$ smaller compartments when we place the small items. Hence, the depth of the binary tree T defining the pseudo-guillotine cutting sequence S' is $O_\varepsilon(1)$ and thus we define at most $O_\varepsilon(1)$ compartments in total. In particular, we applied Lemmas 9 and 10 at most $O_\varepsilon(1)$ times and, therefore, the constructed solution contains at least $(1 - \varepsilon)|OPT| - O_\varepsilon(1)$ items. A refined argument extends this to the weighted case as well, we refer the readers to the full version [37] for a detailed description.

4 Assigning items into compartments

For proving Lemma 5, we need to provide an algorithm that, given a set of compartments \mathcal{C} , computes a solution $ALG \subseteq I$ with $p(ALG) \geq (1 - \varepsilon)p(OPT')$ that can also be placed nicely in \mathcal{C} (where $OPT' \subseteq I$ is the subset of I of maximum profit that can be placed nicely in the compartments in \mathcal{C}).

First, we guess for each box-compartment $B \in \mathcal{C}$ which case of Definition 3 applies, i.e., whether B contains only a single large item, or only horizontal items, or only vertical items, or only small items. For each box-compartment $B \in \mathcal{C}$ for which we guessed that it contains only one large item, we simply guess this item. We can do this deterministically in time $O(n^{|\mathcal{C}|}) = n^{O_\varepsilon(1)}$ for all such box-compartments $B \in \mathcal{C}$.

Then, for assigning the small items, we use a standard reduction to the Generalized Assignment Problem (GAP) [24] for selecting a near-optimal set of small items and an assignment of these items into the corresponding box-compartments. Inside of each box-compartment B we place the items with the Next-Fit-Decreasing-Height algorithm [17] which results in a 2-stage guillotine separable packing for the items inside B .

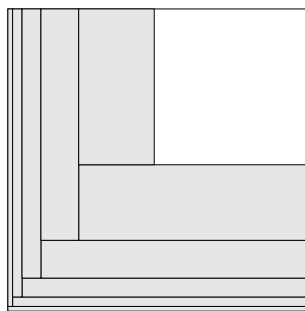
► **Lemma 11** ([37]). *Given a set of box compartments \mathcal{B} such that a set of items $I_{small}^* \subseteq I_{small}$ can be placed non-overlappingly inside \mathcal{B} , in $n^{O_\varepsilon(1)}$ time we can compute a set of items $I'_{small} \subseteq I_{small}$ with $p(I'_{small}) \geq (1 - \varepsilon)p(I_{small}^*)$ and a nice placement of the items in I'_{small} inside \mathcal{B} which is guillotine separable with $O_\varepsilon(1)$ stages.*

Let $\mathcal{C}_{skew} \subseteq \mathcal{C}$ denote the compartments in \mathcal{C} into which skewed items are placed in OPT' (which in particular contains all **L**-compartments in \mathcal{C}). It remains to select a profitable set of items from I_{skew} that can be placed nicely in the compartments in \mathcal{C}_{skew} . For this task, we use a recent algorithm in [26] which is a routine for 2GK which takes as input (in our terminology) a set of box- and **L**-compartments, and also compartments of more general shapes (e.g., with the shapes of a U or a Z). In time $(nN)^{O_\varepsilon(1)}$, it computes a

subset of the input items of maximum total profit, up to a factor of $1 + \varepsilon$, that can be placed non-overlappingly inside the given compartments. In fact, it first partitions the given compartments such that there exists a profitable solution for the smaller compartments inside of which the items are placed nicely (according to our definition). Then it computes a $(1 + \varepsilon)$ -approximation of the most profitable subset of items that can be placed nicely.

In our setting, we can skip the first step since in OPT' the items are already placed nicely inside the compartments \mathcal{C}_{skew} . Hence, we execute directly the second part the algorithm in [26]. In fact, a simpler version of that routine is sufficient since we have only box- and **L**-compartments. The algorithm in [26] can handle also the case where rotations by 90 degree are allowed, and the same holds for the routine in Lemma 11. Thus our result works for the case with rotations as well. We refer to the full version [37] for a complete and self-contained description of this routine, adapted to the guillotine setting. In particular, inside each compartment its solution is guillotine separable with $O_\varepsilon(\log nN)$ stages.

5 Power of stages in guillotine packing



■ **Figure 8** Hard example for Theorem 12.

Our two algorithms compute packings with $O_\varepsilon(\log(nN))$ -stages. This raises the question whether one can obtain $(1 + \varepsilon)$ -approximate solutions with fewer stages. In particular, for the related guillotine 2BP and guillotine 2SP problems there are APTASs whose solutions use $O(1)$ -stage packings [8, 47]. However, we show that in contrast for 2GGK sometimes $\Omega(\log N)$ stages are necessary already for a better approximation ratio than 2, even if there are only skewed items. For a detailed proof, we refer to the full version [37].

► **Theorem 12.** *For any constant $0 < \varepsilon < \frac{1}{2}$, there is a family of instances of 2GGK with only skewed items for which any $(2 - \varepsilon)$ -approximate solution requires $k = \Omega(\varepsilon \log N)$ stages.*

6 Conclusion

Two main open questions are to obtain PTASes for 2GGK and 2GK. We conjecture that the worst-case ratio between the optimal profit of 2GGK and 2GK is $4/3$. If this conjecture is true, then a PTAS for 2GGK will imply a $4/3 + \varepsilon$ -approximation for 2GK, improving the present best approximation guarantee [24].

References

- 1 Fidaa Abed, Parinya Chalermsook, José R. Correa, Andreas Karrenbauer, Pablo Pérez-Lantero, José A. Soto, and Andreas Wiese. On guillotine cutting sequences. In *APPROX*, pages 1–19, 2015.
- 2 Anna Adamaszek and Andreas Wiese. Approximation schemes for maximum weight independent set of rectangles. In *FOCS*, pages 400–409, 2013.
- 3 Anna Adamaszek and Andreas Wiese. A quasi-PTAS for the two-dimensional geometric knapsack problem. In *SODA*, pages 1491–1505, 2015.
- 4 Ramón Alvarez-Valdés, Antonio Parajón, and José Manuel Tamarit. A tabu search algorithm for large-scale guillotine (un) constrained two-dimensional cutting problems. *Computers & Operations Research*, 29(7):925–947, 2002.
- 5 Nikhil Bansal, Alberto Caprara, Klaus Jansen, Lars Prädell, and Maxim Sviridenko. A structural lemma in 2-dimensional packing, and its implications on approximability. In *ISAAC*, pages 77–86, 2009.
- 6 Nikhil Bansal, Jose R Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: inapproximability results and approximation schemes. *Mathematics of Operations Research*, 31:31–49, 2006.
- 7 Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In *SODA*, pages 13–25, 2014.
- 8 Nikhil Bansal, Andrea Lodi, and Maxim Sviridenko. A tale of two dimensional bin packing. In *FOCS*, pages 657–666, 2005.
- 9 J. E. Beasley. An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, 33(1):49–64, 1985.
- 10 István Borgulya. An eda for the 2d knapsack problem with guillotine constraint. *Central European Journal of Operations Research*, 27(2):329–356, 2019.
- 11 Alberto Caprara. Packing 2-dimensional bins in harmony. In *FOCS*, pages 490–499, 2002.
- 12 Alberto Caprara, Andrea Lodi, and Michele Monaci. Fast approximation schemes for two-stage, two-dimensional bin packing. *Mathematics of Operations Research*, 30(1):150–172, 2005.
- 13 Henrik I. Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 24:63–79, 2017.
- 14 Nicos Christofides and Charles Whitlock. An algorithm for two-dimensional cutting problems. *Operations Research*, 25(1):30–44, 1977.
- 15 François Clautiaux, Ruslan Sadykov, François Vanderbeck, and Quentin Viaud. Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem. *Discrete Optimization*, 29:18–44, 2018.
- 16 François Clautiaux, Ruslan Sadykov, François Vanderbeck, and Quentin Viaud. Pattern-based diving heuristics for a two-dimensional guillotine cutting-stock problem with leftovers. *EURO Journal on Computational Optimization*, 7(3):265–297, 2019.
- 17 Edward G. Coffman, Jr, Michael R. Garey, David S. Johnson, and Robert E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9:808–826, 1980.
- 18 Alessandro Di Pieri. Algorithms for two-dimensional guillotine packing problems. Master’s thesis, University of Padova, Italy, 2013.
- 19 Mohammad Dolatabadi, Andrea Lodi, and Michele Monaci. Exact algorithms for the two-dimensional guillotine knapsack. *Computers & Operations Research*, 39(1):48–53, 2012.
- 20 Aleksei V. Fishkin, Olga Gerber, and Klaus Jansen. On efficient weighted rectangle packing with large resources. In *ISAAC*, pages 1039–1050, 2005.
- 21 Aleksei V. Fishkin, Olga Gerber, Klaus Jansen, and Roberto Solis-Oba. Packing weighted rectangles into a square. In *MFCS*, pages 352–363, 2005.

- 22 Fabio Furini, Enrico Malaguti, and Dimitri Thomopulos. Modeling two-dimensional guillotine cutting problems via integer programming. *INFORMS Journal on Computing*, 28(4):736–751, 2016.
- 23 Waldo Gálvez, Fabrizio Grandoni, Afrouz Jabal Ameli, Klaus Jansen, Arindam Khan, and Malin Rau. A tight $(3/2+\epsilon)$ approximation for skewed strip packing. In *APPROX/RANDOM*, pages 44:1–44:18, 2020.
- 24 Waldo Gálvez, Fabrizio Grandoni, Sandy Heydrich, Salvatore Ingala, Arindam Khan, and Andreas Wiese. Approximating geometric knapsack via l-packings. In *FOCS*, pages 260–271, 2017.
- 25 Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, and Arindam Khan. Improved pseudo-polynomial-time approximation for strip packing. In *FSTTCS*, pages 9:1–9:14, 2016.
- 26 Waldo Gálvez, Fabrizio Grandoni, Arindam Khan, Diego Ramirez-Romero, and Andreas Wiese. Improved approximation algorithms for 2-dimensional knapsack: Packing into multiple l-shapes, spirals and more. In *To appear in SoCG*, 2021.
- 27 P. C. Gilmore and Ralph E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations research*, 13(1):94–120, 1965.
- 28 Fabrizio Grandoni, Tobias Mömke, Andreas Wiese, and Hang Zhou. A $(5/3 + \epsilon)$ -approximation for unsplittable flow on a path: placing small tasks into boxes. In *STOC*, pages 607–619, 2018.
- 29 Eleni Hadjiconstantinou and Nicos Christofides. An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of OR*, 83(1):39–56, 1995.
- 30 Rolf Harren, Klaus Jansen, Lars Prädell, and Rob van Stee. A $(5/3 + \epsilon)$ -approximation for strip packing. *Computational Geometry*, 47(2):248–267, 2014.
- 31 Sandy Heydrich and Andreas Wiese. Faster approximation schemes for the two-dimensional knapsack problem. In *SODA*, pages 79–98, 2017.
- 32 Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.
- 33 Klaus Jansen and Roberto Solis-Oba. New approximability results for 2-dimensional packing problems. In *MFCS*, pages 103–114, 2007.
- 34 Klaus Jansen and Roberto Solis-Oba. A polynomial time approximation scheme for the square packing problem. In *IPCO*, pages 184–198, 2008.
- 35 Klaus Jansen and Guochuan Zhang. On rectangle packing: maximizing benefits. In *SODA*, pages 204–213, 2004.
- 36 Claire Kenyon and Eric Rémila. A near-optimal solution to a two-dimensional cutting stock problem. *Mathematics of Operations Research*, 25(4):645–656, 2000.
- 37 Arindam Khan, Arnab Maiti, Amatya Sharma, and Andreas Wiese. On guillotine separable packings for the two-dimensional geometric knapsack problem, 2021. [arXiv:2103.09735](https://arxiv.org/abs/2103.09735).
- 38 Arindam Khan and Madhusudhan Reddy Pittu. On guillotine separability of squares and rectangles. In *APPROX/RANDOM*, pages 47:1–47:22, 2020.
- 39 Arindam Khan, Eklavya Sharma, and K. V. N. Sreenivas. Approximation algorithms for generalized multidimensional knapsack. *CoRR*, abs/2102.05854, 2021.
- 40 Joseph Y. T. Leung, Tommy W. Tam, C. S. Wong, Gilbert H. Young, and Francis Y. L. Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3):271–275, 1990.
- 41 Andrea Lodi, Michele Monaci, and Enrico Pietrobuoni. Partial enumeration algorithms for two-dimensional bin packing problem with guillotine constraints. *Discrete Applied Mathematics*, 217:40–47, 2017.
- 42 Tobias Mömke and Andreas Wiese. Breaking the barrier of 2 for the storage allocation problem. In *ICALP*, pages 86:1–86:19, 2020.
- 43 János Pach and Gábor Tardos. Cutting glass. In *SoCG*, pages 360–369, 2000.
- 44 V. Parada, R. Munoz, and A. Gomes. A hybrid genetic algorithm for the two-dimensional cutting problem. *Evolutionary algorithms in management applications*. Springer, Berlin, pages 183–196, 1995.

- 45 Deval Patel, Arindam Khan, and Anand Louis. Group fairness for knapsack problems. In *To appear in AAMAS*, 2021.
- 46 Enrico Pietrobuoni. *Two-dimensional bin packing problem with guillotine restrictions*. PhD thesis, University of Bologna, Italy, 2015.
- 47 Steven S. Seiden and Gerhard J. Woeginger. The two-dimensional cutting stock problem revisited. *Mathematical Programming*, 102(3):519–530, 2005.
- 48 A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing*, 26(2):401–409, 1997.
- 49 Paul E. Sweeney and Elizabeth Ridenour Paternoster. Cutting and packing problems: a categorized, application-orientated research bibliography. *Journal of the Operational Research Society*, 43(7):691–706, 1992.
- 50 K. V. Viswanathan and A. Bagchi. An exact best-first search procedure for the constrained rectangular guillotine knapsack problem. In *AAAI*, pages 145–149, 1988.
- 51 Lijun Wei and Andrew Lim. A bidirectional building approach for the 2d constrained guillotine knapsack packing problem. *European Journal of Operational Research*, 242(1):63–71, 2015.