






Improved Approximation Algorithms for Three-Dimensional Knapsack

Klaus Jansen   

Kiel University, Germany

Debajyoti Kar   



Indian Institute of Science, Bengaluru, India

Arindam Khan   

Indian Institute of Science, Bengaluru, India

K. V. N. Sreenivas   

Indian Institute of Science, Bengaluru, India

Malte Tutas  

Kiel University, Germany

Abstract

We study the three-dimensional Knapsack (3DK) problem, in which we are given a set of axis-aligned cuboids with associated profits and an axis-aligned cube knapsack. The objective is to find a non-overlapping axis-aligned packing (by translation) of the maximum profit subset of cuboids into the cube. The previous best approximation algorithm is due to Diedrich, Harren, Jansen, Thöle, and Thomas (2008), who gave a $(7 + \varepsilon)$ -approximation algorithm for 3DK and a $(5 + \varepsilon)$ -approximation algorithm for the variant when the items can be rotated by 90 degrees around any axis, for any constant $\varepsilon > 0$. Chlebík and Chlebíková (2009) showed that the problem does not admit an asymptotic polynomial-time approximation scheme.

We provide an improved polynomial-time $(139/29 + \varepsilon) \approx 4.794$ -approximation algorithm for 3DK and $(30/7 + \varepsilon) \approx 4.286$ -approximation when rotations by 90 degrees are allowed. We also provide improved approximation algorithms for several variants such as the cardinality case (when all items have the same profit) and uniform profit-density case (when the profit of an item is equal to its volume). Our key technical contribution is *container packing* – a structured packing in 3D such that all items are assigned into a constant number of containers, and each container is packed using a specific strategy based on its type. We first show the existence of highly profitable container packings. Thereafter, we show that one can find near-optimal container packing efficiently using a variant of the Generalized Assignment Problem (GAP).

2012 ACM Subject Classification Theory of computation → Packing and covering problems

Keywords and phrases Approximation Algorithms, Hyperrectangle Packing, Multidimensional Knapsack, Three-dimensional Packing

Digital Object Identifier 10.4230/LIPIcs.SoCG.2025.60

Related Version *Full Version:* <https://arxiv.org/abs/2503.19365> [16]

Funding *Klaus Jansen:* Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 453769249.

Debajyoti Kar: Google PhD Fellowship.

Arindam Khan: Google India Research Award, SERB Core Research Grant (CRG/2022/001176) on “Optimization under Intractability and Uncertainty”, Ittiam Systems CSR grant, and the Walmart Center for Tech Excellence at IISc (CSR Grant WMGT-23-0001).

K. V. N. Sreenivas: Google PhD Fellowship.

Malte Tutas: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 453769249.



© Klaus Jansen, Debajyoti Kar, Arindam Khan, K. V. N. Sreenivas, and Malte Tutas; licensed under Creative Commons License CC-BY 4.0

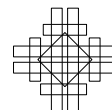
41st International Symposium on Computational Geometry (SoCG 2025).

Editors: Oswin Aichholzer and Haitao Wang; Article No. 60; pp. 60:1–60:18

Leibniz International Proceedings in Informatics



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



Acknowledgements We are grateful to Lars Prädél for helping us with some details of the 3D Strip Packing result. We are also thankful to Marvin Lira for initial discussions. We also thank the anonymous reviewers for their valuable comments.

1 Introduction

Three-dimensional Knapsack (3DK) is a fundamental problem in computational geometry, operations research, and approximation algorithms. In 3DK, we are given a set of items which are axis-aligned cuboids (defined by their width, depth, and height) with associated profits, and a three-dimensional (3D) knapsack which is an axis-aligned unit cube. The goal is to find a nonoverlapping axis-aligned packing (by translation) of a subset of items into the knapsack such that the profit of the packed items is maximized.

In 1965, Gilmore and Gomory [13] introduced the 3D Knapsack problem – motivated by cutting stock problems (e.g., cutting up of graphite blocks for anodes). They provided heuristics for the special case when the items need to be cut along axis-aligned planes. Since then, the problem and its several variants (under practical constraints) have been well-studied in operations research – typically from the viewpoints of meta-heuristics, MILP, tree search, machine learning methods, and greedy algorithms [5, 25]. In the last decade, 3DK has become a central problem in logistics to increase operational efficiency by efficiently utilizing the space in various domains, such as airline cargo management, warehouse management, and robotic container loading [1]. The prominence of this problem in the industry is evidenced by the comprehensive survey on 3D packing by Ali, Ramos, Carravilla, and Oliveira [1], who have mentioned hundreds of papers in recent years.

Packing problems are fundamental and popular among mathematicians (e.g., see Kepler’s sphere packing conjecture [23] and *Soma cube* puzzle [14]). Surprisingly, 3DK is relatively less studied in computational geometry and approximation algorithms. Only in 2008, Diedrich, Harren, Jansen, Thöle, and Thomas [10] provided the first provable approximation guarantees for the problem. They first gave a simple polynomial-time $(9 + \varepsilon)$ -approximation algorithms for any constant $\varepsilon > 0$. Thereafter, they gave a more sophisticated $(7 + \varepsilon)$ -approximation algorithm for 3DK. For the case when the items can be rotated by 90 degrees around any axis, they provided a $(5 + \varepsilon)$ -approximation algorithm. Afterward, Chlebík and Chlebíková [7] showed that the problem admits no asymptotic polynomial-time approximation schemes (APTAS) even for the cardinality case (when each item has the same profit). Lu, Chen, and Cha [24] showed that even packing cubes into a cube is strongly NP-hard.

There has been progress in some special cases. When all items are d -dimensional hypercubes, Harren [15] gave a $(1 + 2^{-d} + \varepsilon)$ -approximation algorithm. Recently, Jansen, Khan, Lira, and Sreenivas [17] gave a PTAS for this special case. Afterwards, Buchem, Deuker, and Wiese [4] obtained an EPTAS. Furthermore, they gave dynamic algorithms with polylogarithmic query and update times, matching the same approximation guarantees.

For the 2D variant when all items are rectangles (2DK), Jansen and Zhang [20] gave a $(2 + \varepsilon)$ -approximation algorithm. Gálvez, Grandoni, Ingala, Heydrich, Khan, and Wiese [11] gave an improved 1.89-approximation algorithm. Later, Gálvez, Grandoni, Khan, Ramírez-Romero, and Wiese [12] gave a pseudopolynomial-time $4/3$ -approximation algorithm. For the variant of *guillotine packing* (where items need to be packed such that they can be separated by a series of end-to-end cuts), Khan, Maiti, Sharma, and Wiese [21] gave a pseudopolynomial-time approximation scheme (PPTAS) for the problem.

Compared to the classical (1D) Knapsack, multidimensional Knapsack is much harder as there are different types of items and they can interact in a complicated way. For example, 2D items can be big (large in both dimensions), small (tiny in both dimensions), vertical (large in

height, small in width), or horizontal (large in width, small in height). Most approximation algorithms for 2DK show the existence of a highly profitable structured packing where the knapsack is partitioned into a constant number of boxes such that each box contains only items of one type and can be packed easily by simple greedy algorithms. Large boxes contain a single large item, vertical (resp. horizontal) boxes contain only vertical (resp. horizontal) items (thus, one can essentially ignore the long dimension of the item, and the box becomes a 1D knapsack), and small boxes contain small items (one can consider the area of an item as a proxy and use 1D knapsack solution). However, this is difficult for 3D packing. As pointed out by Bansal, Correa, Kenyon, and Sviridenko [3], one of the obstacles to generalize methods from rectangle packing to cuboid packing is that “*the interaction of say 3-dimensional rectangles which are long in one direction but short in the other two seems much more complicated than in the two-dimensional case.*”

In 2008, the authors in [10] mentioned that “*it is of interest whether here (for 3DK) an algorithm with ratio $(6 + \varepsilon)$ or less exists.*” However, despite significant progress in the 2D case, there has been no improved approximation algorithm for 3DK in almost two decades.

1.1 Our results

In this paper, we introduce *container packing* for 3D Knapsack. We show that we can divide the 3D knapsack into $O(1)$ number of regions (called *containers*) such that the dimensions of these containers come from a polynomial-sized set. For each container C we define its capacity $\text{cap}(C) \in \mathbb{Q}_{\geq 0}$ and for each item i we define its size in C to be $f_C(i) \in \mathbb{Q}_{\geq 0}$. We also associate an algorithm \mathcal{A}_C for the container. We require \mathcal{A}_C, f_C such that for any C and any set of items T satisfying the packing constraint $\sum_{i \in T} f_C(i) \leq \text{cap}(C)$, almost all items in T (barring a small profit subset) can be packed into C by algorithm \mathcal{A}_C . The assignment of items into containers thus becomes a variant of the Multiple Knapsack problem (however, the knapsacks can have different capacities and items can have sizes specific to the knapsacks) – also a special case of Generalized Assignment Problem (GAP). For $O(1)$ knapsacks, there exists a PTAS for GAP [11]. After the assignment of items to the container C , the corresponding algorithm \mathcal{A}_C ensures that the items admit a feasible profitable packing.

Container packing is a powerful and general technique. For example, by choosing appropriate functions f_C , our containers can handle complicated packings, beyond simple greedy algorithms. For example, we use six different types of containers in this work (see Figures 4 and 8). Apart from the Stack containers and Volume containers (similar to 2D packing), we introduce Area containers (packed using a variant of Next-Fit-Decreasing-Height (NFDH) algorithm [9]), Steinberg containers (where different types of items are packed together using our new volume-based packing algorithm **3D-Vol-Pack** – it packs items in layers where each layer is packed using either a greedy algorithm or a 2D packing algorithm called Steinberg’s algorithm [26]), and L-Containers (where two different types of stacks are packed together when rotations are allowed). Note that Steinberg container or L-Container goes beyond simple stack or shelf-based algorithms and equips us to handle more complicated packings of different types of items. Finally, we show that given an optimal packing, we can modify it to some *container packing* having significant profit. The functions f_C can be considered analogous to the *weighting functions* in bin packing [9], and we expect that more complicated packing algorithms can be incorporated into this framework by suitably defining f_C .

Using this, we obtain improved approximation guarantees for 3DK (see Table 1). In particular, we obtain $(139/29 + \varepsilon)$ -approximation for 3DK and $(30/7 + \varepsilon)$ -approximation for 3DKR (when rotations by 90 degrees are allowed around all axes). For the cardinality case

■ **Table 1** Summary of our results.

	General case	Cardinality case	Profit = Volume
Without rotations	$\frac{139}{29} + \varepsilon < 4.794$	$\frac{17}{4} + \varepsilon = 4.25 + \varepsilon$	$4 + \varepsilon$
With rotations	$\frac{30}{7} + \varepsilon < 4.286$	$\frac{24}{7} + \varepsilon < 3.429$	$3 + \varepsilon$

(when all items have the same profit), we obtain $17/4 + \varepsilon$ and $24/7 + \varepsilon$, respectively, for the cases without and with rotations. Finally, in the case when the profit of an item is equal to its volume (also called uniform profit-density), we obtain approximation guarantees of $4 + \varepsilon$ and $3 + \varepsilon$, without and with rotations, respectively.

Due to space limitations, some parts of the proofs are omitted in this extended abstract. For missing details, we refer the reader to the full version [16].

1.2 Related work

Two other related problems are 3D Bin Packing (BP) and Strip Packing (SP). In 3D BP, given a set of cuboids and unit cubes as bins, the goal is to find a nonoverlapping axis-aligned packing of all items into the minimum number of bins. In 3D SP, given a set of cuboids, the goal is to pack them into a single 3D rectangular box of unlimited height such that the height of the packing is minimized. For 3D BP and 3D SP, the best-known (asymptotic) approximation ratios are $T_\infty^2 \approx 2.86$ [6] and $3/2 + \varepsilon$ [19], respectively. For more details on related problems, we refer the readers to surveys [2, 8] on approximation and online algorithms for multidimensional packing.

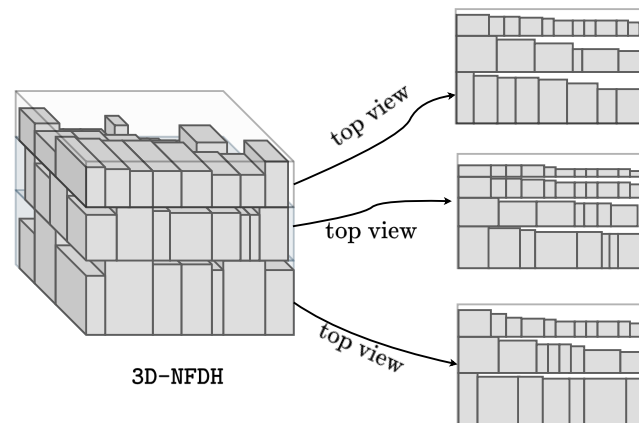
2 Notation and Preliminaries

We are given a set of n items $I = \{1, 2, \dots, n\}$. Each item $i \in I$ is a cuboid with width $w_i \in (0, 1]$, depth $d_i \in (0, 1]$, height $h_i \in (0, 1]$, and an associated profit $p_i \in \mathbb{Q}$. The volume of an item i is $v(i) = w_i \times d_i \times h_i$, and its base area is defined as $w_i d_i$. We are also given a 3D unit cube knapsack $K := [0, 1] \times [0, 1] \times [0, 1]$. If an item i is placed (by translation) in an axis-aligned way at (i_x, i_y, i_z) then it occupies the region: $[i_x, i_x + w_i] \times [i_y, i_y + d_i] \times [i_z, i_z + h_i]$. For a feasible packing we need $i_x \in [0, 1 - w_i]$, $i_y \in [0, 1 - d_i]$, $i_z \in [0, 1 - h_i]$. In a packing, two items are nonoverlapping if their interiors are disjoint. Our goal is to find a nonoverlapping packing of a subset of these items that maximizes the total profit. In 3DK, we allow only translation, whereas 3DKR allows translation and rotation by 90 degrees around all axes. For any set of items T , we shall let $p(T)$, $h(T)$ and $v(T)$ denote the total profit, sum of heights, and volume of T , respectively. Let OPT denote an optimal packing and $\text{opt} := p(\text{OPT})$.

2.1 Packing subroutines

We now describe a few procedures that we repeatedly use throughout the paper to pack items. For a cuboidal box B , let w_B, d_B, h_B denote the width, depth, and height of B , respectively. The volume of the box is then $v(B) = w_B \times d_B \times h_B$.

Next-Fit-Decreasing-Height (NFDH) Algorithm [9]. NFDH is a shelf-based packing algorithm for packing rectangles. 3D-NFDH is its generalization in three dimensions (see Figure 1). The following lemmas summarize the guarantees of these algorithms.



■ **Figure 1** 3D-NFDH with three layers. Each layer is packed using (two-dimensional) NFDH.

► **Lemma 1** ([11]). *We are given a rectangular box of length ℓ (horizontal dimension) and breadth b (vertical dimension), and a set S of n (2D) rectangles, where each $i \in S$ has length $\ell_i \leq \varepsilon\ell$ and breadth $b_i \leq \varepsilon b$. If $\sum_{i \in S} \ell_i b_i \leq (1 - 2\varepsilon)\ell b$, then the whole set S can be packed into the box using NFDH in $O(n \log n)$ time.*

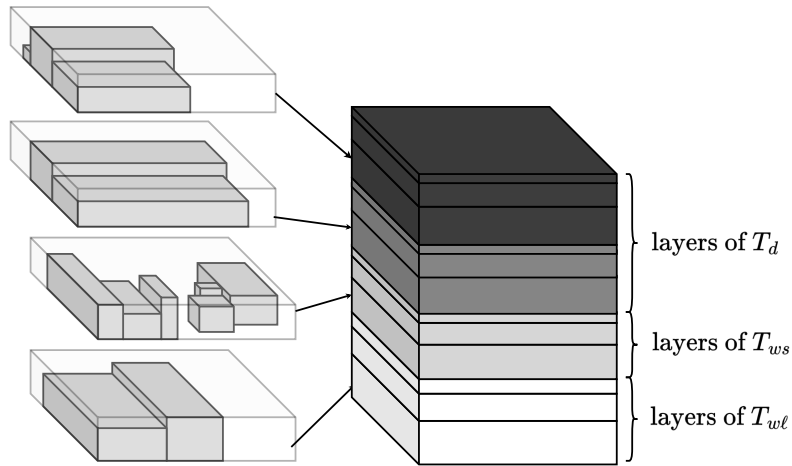
► **Lemma 2.** *We are given a cuboidal box B , and a set T of n items where each $i \in T$ satisfies $w_i \leq \varepsilon w_B$, $d_i \leq \varepsilon d_B$ and $h_i \leq \varepsilon h_B$. If $v(T) \leq (1 - 3\varepsilon)v(B)$, then the whole set T can be packed into B using 3D-NFDH in $O(n \log^2 n)$ time.*

Steinberg's Algorithm [26]. Steinberg's Algorithm is a commonly used 2D packing algorithm with the following performance guarantee.

► **Lemma 3** ([26]). *We are given a set S of n rectangles and a rectangular box of size $\ell \times b$. Let $\ell_{\max} \leq \ell$ and $b_{\max} \leq b$ be the maximum length and maximum breadth among the rectangles in S , respectively, and let $a(S)$ be the total area of the rectangles in S . Also, we denote $x_+ := \max(x, 0)$. If $2a(S) \leq \ell b - (2\ell_{\max} - \ell)_+(2b_{\max} - b)_+$, then there is an algorithm that can pack the whole set S into the rectangular box in $O\left(\frac{n \log^2 n}{\log \log n}\right)$ time.*

We utilize the above algorithm to pack items in layers inside a box. Below, we present two algorithms 3D-Vol-Pack and 3DR-Vol-Pack for packing 3D items for the cases without and with rotations, respectively. As Steinberg's algorithm is used as a black box in many 2D packing problems, we believe our volume-based bounds and algorithms should find usage in other 3D packing problems. In both cases, we are given a box B with height h_B , width w_B , and depth d_B , and a set of items T .

3D-Vol-Pack. We assume that each $i \in T$ satisfies either $w_i \leq w_B/2$ or $d_i \leq d_B/2$. Let T_w be the items of T having width at most $w_B/2$, and let $T_d = T \setminus T_w$. Then, similar to [10], we further classify T_w as follows: let $T_{w\ell} \subseteq T_w$ be the items whose base area (area of the bottom face) exceeds $\frac{1}{6}w_B d_B$, and let $T_{ws} = T_w \setminus T_{w\ell}$. Next, we sort the sets $T_{w\ell}$ and T_{ws} in non-increasing order of heights. We group the items of $T_{w\ell}$ in pairs (except possibly the last item), and pack each pair in a single layer. See Figure 2. For packing items in T_{ws} , we group them into maximal collections of total base area not exceeding $\frac{1}{2}w_B d_B$, and pack each collection in a single layer using Steinberg's algorithm (Lemma 3). Analogously, we obtain a



■ **Figure 2** An example packing by Algorithm 3D-Vol-Pack.

packing of the items in T_d in layers. Finally, we stack the layers one above the other inside the box B as long as the height of the box is not exceeded. The following lemma provides a guarantee when the item heights are small compared to the box height.

► **Lemma 4.** *Given a box B , and a set T of n items where each $i \in T$ satisfies $w_i \leq w_B$, $d_i \leq d_B$ and $h_i \leq \varepsilon h_B$. Further, either $w_i \leq w_B/2$ or $d_i \leq d_B/2$ holds for every $i \in T$. If $v(T) \leq (\frac{1}{3} - 2\varepsilon)v(B)$, then 3D-Vol-Pack packs the whole set T into B in $O\left(\frac{n \log^2 n}{\log \log n}\right)$ time.*

Next, we give an algorithm when the items can be rotated by 90 degrees about any axis.

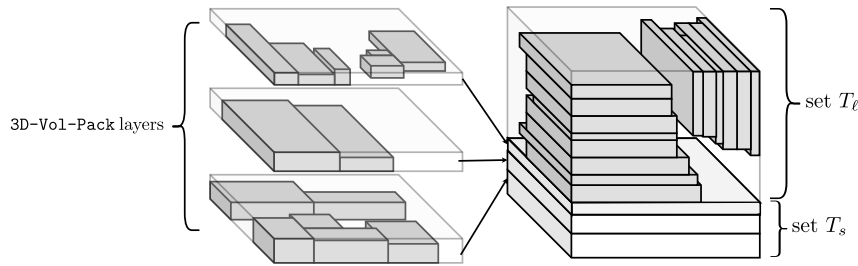
3DR-Vol-Pack. Let $T_\ell = \{i \in T \mid w_i > w_B/2 \text{ and } d_i > d_B/2\}$, and let $T_s = T \setminus T_\ell$. First, we pack T_s in layers using 3D-Vol-Pack. Next, we sort the items of T_ℓ in non-increasing order of their widths and pack them in layers, one in each layer, touching the left face of the box, as long as the height of the box is not exceeded. Following this, we rotate the remaining items of T_ℓ about the depth dimension so that their widths and heights are interchanged, and as long as possible, pack them next to each other starting from the right face of the box, such that each item touches the top face of the box (see Figure 3).

The next lemma states that when B is a cube and when the height of each item is very small compared to box B , we can pack a significant volume.

► **Lemma 5.** *Given a cubical box B of side length w , and a set T of n items where for each $i \in T$, there exists an orientation so that $w_i \leq w$, $d_i \leq w$ and $h_i \leq \varepsilon^2 w$ holds. If $v(T) \leq (\frac{7}{24} - 5\varepsilon)v(B)$, then T can be packed into B by 3DR-Vol-Pack in $O\left(\frac{n \log^2 n}{\log \log n}\right)$ time.*

3 Container Packing

In this section, we first define container packing and describe the containers used in our work. Then, we state our main structural result which shows that there exist container packings with high profit that can be found in polynomial time.



■ **Figure 3** An example packing by Algorithm 3DR-Vol-Pack.

► **Definition 6** (Container Packing). A packing of a set of items $I' \subseteq I$ is said to be a *Container Packing* if

- the knapsack can be partitioned into a collection \mathcal{C} of non-overlapping regions called *containers* and some empty spaces. Each container $C \in \mathcal{C}$ has an associated capacity $\text{cap}(C)$, a function $f_C: I \rightarrow \mathbb{Q}_{\geq 0}$, and a polynomial-time algorithm \mathcal{A}_C ,
- there is a function $g: I' \rightarrow \mathcal{C}$, such that for each $C \in \mathcal{C}$, items in $g^{-1}(C)$ are packed into C using algorithm \mathcal{A}_C ,
- for any $C \in \mathcal{C}$ and $T \subseteq I$ such that $\sum_{i \in T} f_C(i) \leq \text{cap}(C)$, for any constant $\varepsilon > 0$, there exists a polynomial-time computable set $T' \subseteq T$ with $p(T') \geq (1 - O(\varepsilon))p(T)$ such that T' can be packed into C by the algorithm \mathcal{A}_C .

A Container Packing is said to be *guessable* if

- the number of containers inside the knapsack is bounded by a constant $O_\varepsilon(1)$,¹
- the number of distinct possible types of containers (defined by their sizes along each dimension) is bounded by $n^{O_\varepsilon(1)}$, a polynomial in n ,
- whether a given collection of $O_\varepsilon(1)$ containers can be placed non-overlappingly inside the knapsack, can be checked in $n^{O_\varepsilon(1)}$ (polynomial) time.

We shall call a container *guessable* if it comes from some polynomially-sized set of distinct container types. Though, in this paper we only use cuboids as containers, the idea of container packing can be generalized to other shapes in 3D or higher dimensions.

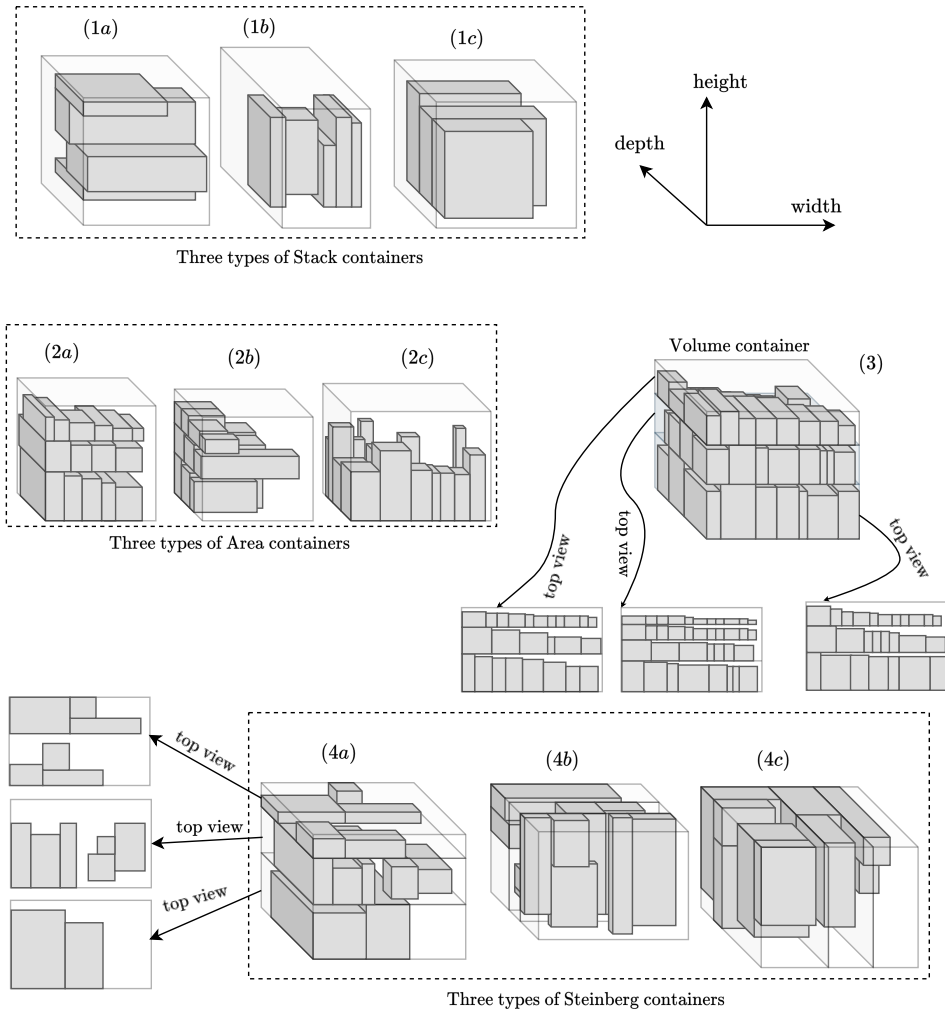
► **Theorem 7.** *There is a PTAS for maximum profit guessable container packing.*

Proof. We consider all feasible choices of containers that fit into the 3D knapsack, and for each such choice, we create an instance of GAP with one (1D) knapsack per container, where the (1D) knapsack corresponding to a container C has capacity $\text{cap}(C)$ and the size of an item i for the (1D) knapsack is given by $f_C(i)$. The profit of an item for any (1D) knapsack is set to be the same as the actual profit of the item. Using the PTAS for GAP with $O(1)$ knapsacks [11], we obtain a feasible assignment of items into the containers, and finally for each container C , we use the algorithm \mathcal{A}_C to pack a subset of the items assigned to it, by losing only an $O(\varepsilon)$ -fraction of profit. ◀

3.1 Classification of containers

We now describe the various types of containers that we use to pack items. For a container C , its width, depth, and height are denoted by w_C, d_C , and h_C , respectively, and its volume

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



■ **Figure 4** Different kinds of containers that we use. We have one type of Volume container and three types of Stack, Area, and Steinberg containers (one for each dimension).

$v(C) := w_C \times d_C \times h_C$. For each container C , we specify its capacity $\text{cap}(C)$, the function f_C , and the packing algorithm \mathcal{A}_C . Let T denote the set of items assigned to C such that $\sum_{i \in T} f_C(i) \leq \text{cap}(C)$. We suggest the reader to follow the descriptions alongside Figure 4.

- **Stack containers:** These containers pack items in layers, with one item in each layer. We consider a container C that stacks items along the height. See container (1a) in Figure 4. The capacity $\text{cap}(C)$ of such a container is set to its height h_C , and for any item i , $f_C(i) := h_i$ if $w_i \leq w_C$ and $d_i \leq d_C$, and $f_C(i) := \infty$ otherwise. \mathcal{A}_C just stacks the items of set T in the container C one above the other. Analogously containers (1b) and (1c) show containers that stack along width and depth, respectively.
- **Area containers:** Inside these containers, the items are packed using the NFDH algorithm, by ignoring one of the dimensions. We describe Area containers that use NFDH along the front face of the container (see container (2a)). For such a container C , we set $\text{cap}(C) := w_C h_C$, and for any item i , we set $f_C(i) := w_i h_i$ if $w_i \leq \varepsilon w_C$, $h_i \leq \varepsilon h_C$ and $d_i \leq d_C$, and $f_C(i) := \infty$ otherwise.

\mathcal{A}_C first sorts the items of set T in non-increasing order of profit/front area ratio. Then, we select the largest prefix whose total front area does not exceed $(1 - 2\varepsilon)w_C h_C$. We pack this prefix using NFDH on the front face. Analogously, container (2b) and (2c) show packing when NFDH is used on the left and bottom face of C , respectively.

- **Volume containers:** The items are packed using 3D-NFDH inside these containers (see container (3)). The capacity of such a container C is set to be its volume $v(C)$, and $f_C(i) := v(i)$ if $w_i \leq \varepsilon w_C$, $d_i \leq \varepsilon d_C$ and $h_i \leq \varepsilon h_C$, and $f_C(i) := \infty$ otherwise.

\mathcal{A}_C sorts the items of T in non-decreasing order of profit/volume and selects the largest prefix with volume not exceeding $(1 - 3\varepsilon)w_C h_C d_C$. We pack this prefix using 3D-NFDH.

- **Steinberg containers:** These containers pack items in layers using 3D-Vol-Pack. We describe Steinberg containers that pack items in layers along the height (analogous for other cases). We set $\text{cap}(C) := v(C)/3$. For any item i , we set $f_C(i) := v(i)$ if $h_i \leq \varepsilon h_C$, and either $w_i \leq w_C/2$ or $d_i \leq d_C/2$ holds. For all other items, we set $f_C(i) := \infty$.

\mathcal{A}_C first sorts the items in T in non-increasing order of profit/volume ratio. Then, we select the largest prefix whose volume does not exceed $(\frac{1}{3} - 2\varepsilon)v(C)$, and pack this prefix using 3D-Vol-Pack. Analogously, container (4b) and (4c) show packing when layers are stacked along the depth and width, respectively.

3.2 Our Structural Lemma

Let opt_{gs} be the maximum profit of a guessable container packing. The main structural result of our paper is the following.

- **Lemma 8.** *For any $\varepsilon > 0$, the optimal profit opt and the optimal profit of a guessable container packing opt_{gs} satisfies $\text{opt} \leq (\frac{139}{29} + \varepsilon) \text{opt}_{\text{gs}}$.*

Together with Theorem 7, this gives us the following theorem.

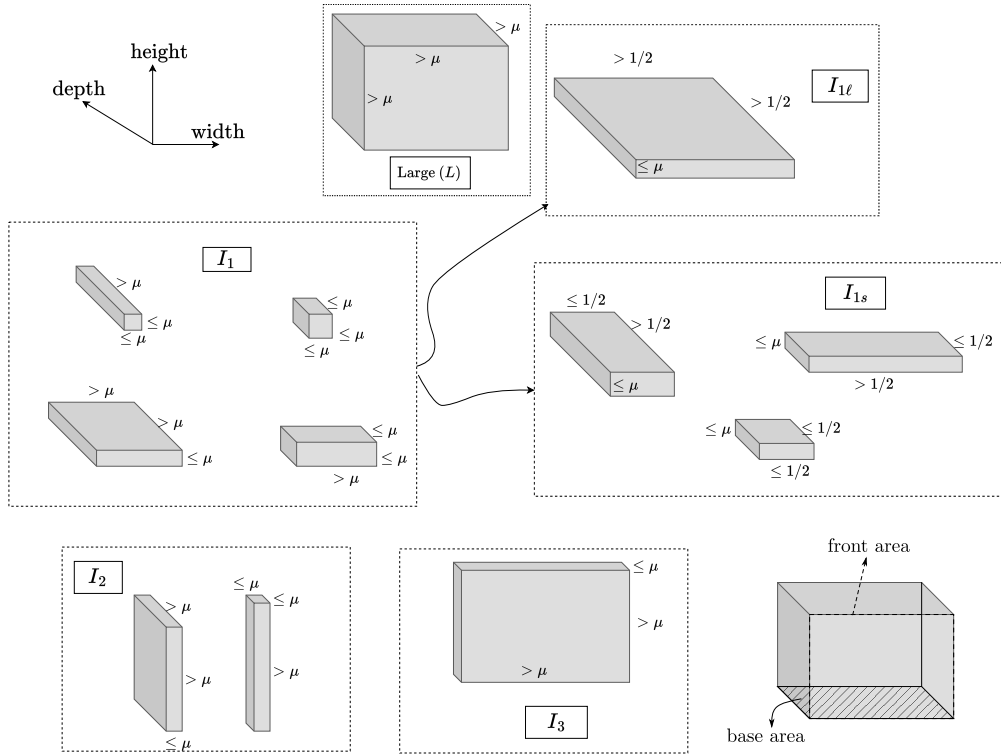
- **Theorem 9.** *For any constant $\varepsilon > 0$, there is a polynomial-time $(\frac{139}{29} + \varepsilon)$ -approximation algorithm for 3DK.*

We prove Lemma 8 in Section 4 by establishing several lower bounds on opt_{gs} . Specifically, we present five different ways to restructure OPT into a guessable container packing. Our first lower bound is based on a structural result of the 3D Strip Packing problem due to [19]. The second lower bound is obtained by restructuring the packing of items in $\text{OPT} \cap (I_{1\ell} \cup L)$ (or similarly $\text{OPT} \cap (I_{2\ell} \cup L)$ or $\text{OPT} \cap (I_{3\ell} \cup L)$). The third and fourth lower bounds are based on volume-based arguments, using our simple packing algorithm 3D-Vol-Pack. Our final lower bound is based on a structural result from the 2D Knapsack problem [11].

4 Proof of Structural Lemma

Now we prove the structural lemma. First, we classify the items.

Classification of items. Let $\varepsilon > 0$ be an accuracy parameter. Let μ be a sufficiently small constant depending on ε (setting $\mu := \varepsilon^{2^{1/\varepsilon^2}}$ suffices). We now classify the input items based on their dimensions. Let $L \subseteq I$ be the set of items whose width, depth and height all exceed μ . Let $I_1 \subseteq I$ be the set of items having height at most μ . Similarly, let $I_2 \subseteq I \setminus I_1$ be those items whose widths are at most μ , and $I_3 := I \setminus (L \cup I_1 \cup I_2)$ be the remaining items. Thus each item in I_3 has depth at most μ . Let $I_{1\ell}$ consist of the items in I_1 whose width and depth are both more than $1/2$, and let $I_{1s} = I_1 \setminus I_{1\ell}$. See Figure 5. The sets $I_{2\ell}, I_{2s}, I_{3\ell}, I_{3s}$ are defined analogously.



■ **Figure 5** Classification of input items. Front area and base area are shown at the bottom right.

Let $\text{opt}_i := p(\text{OPT} \cap I_i)$ for $i \in [3]$, and $\text{opt}_L := p(\text{OPT} \cap L)$, so that $\text{opt} = \text{opt}_1 + \text{opt}_2 + \text{opt}_3 + \text{opt}_L$. Let $\text{opt}_{1\ell} := p(\text{OPT} \cap I_{1\ell})$ and $\text{opt}_{1s} := p(\text{OPT} \cap I_{1s})$, so that $\text{opt}_{1\ell} + \text{opt}_{1s} = \text{opt}_1$. The quantities $\text{opt}_{2\ell}, \text{opt}_{2s}$ and $\text{opt}_{3\ell}, \text{opt}_{3s}$ are defined analogously for the sets I_2 and I_3 , respectively.

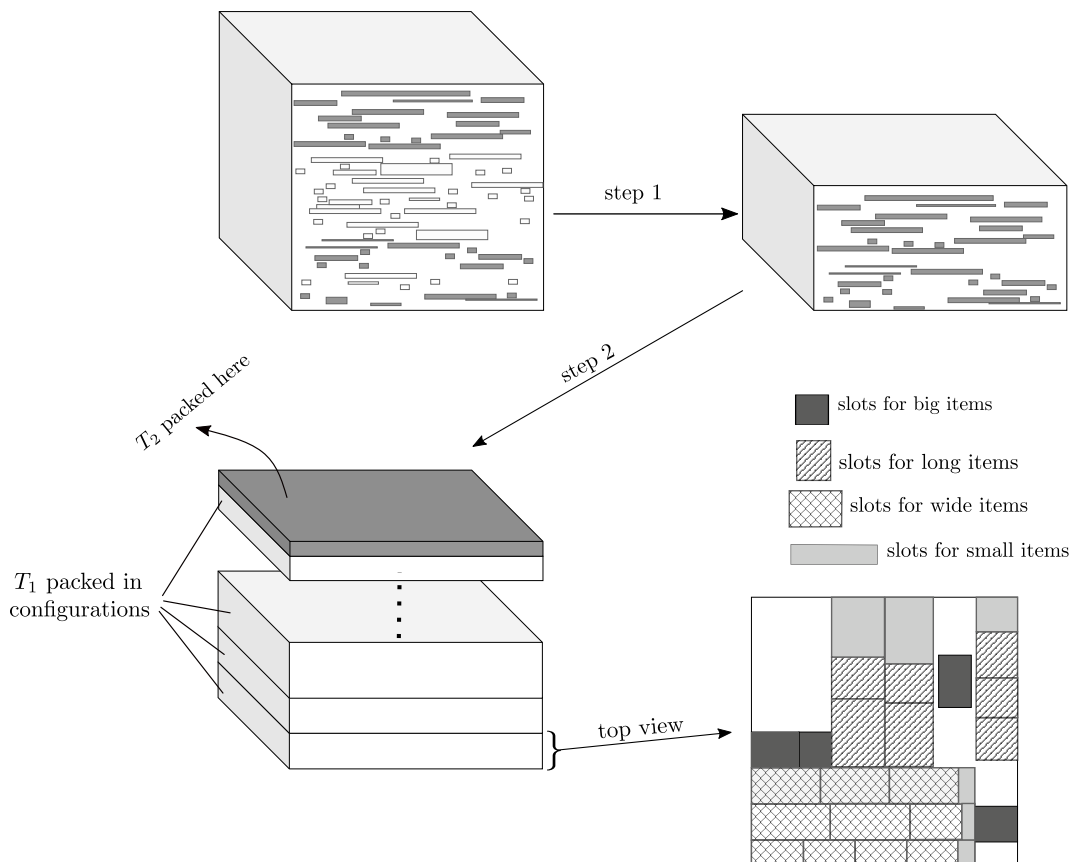
Finally, let $v_1 := v(\text{OPT} \cap I_1)$, and define v_2, v_3 analogously. Clearly, v_1, v_2, v_3 sum up to at most the volume of the knapsack which is 1. We define $h_{1\ell} := h(\text{OPT} \cap I_{1\ell})$ and $v_{1s} := v(\text{OPT} \cap I_{1s})$. The quantities v_{2s}, v_{3s} are defined analogously.

4.1 First lower bound on opt_{gs}

We consider the packing of the items in $\text{OPT} \cap I_1$ (i.e., items with height at most μ) whose total profit is opt_1 . We build on the structure derived from the 3D Strip Packing algorithm of Jansen and Prädél [19]. However, we need several refinements to transform the packing of [19] into a guessable container packing.

First, using standard arguments, we discard a set of (medium) items (items whose widths or depths lie in the range $[\delta^{10}, \delta)$ for a suitably chosen δ) at a $(1 + \varepsilon)$ -factor loss in the approximation ratio. Then the items are divided into four classes depending on their dimensions. An item i is big if both $w_i \geq \delta$ and $d_i \geq \delta$, wide if $w_i \geq \delta$ and $d_i < \delta^{10}$, long if $w_i < \delta^{10}$ and $d_i \geq \delta$, and small if both $w_i < \delta^{10}$ and $d_i < \delta^{10}$. The following lemma is obtained by an adaptation of the result of [19] (see Figure 6).

- ▶ **Lemma 10.** *There exist disjoint sets $T_1, T_2 \subseteq OPT \cap I_1$ having a profit of at least $(\frac{2}{3} - O(\varepsilon)) \text{opt}_1$, such that*
 - *Items of T_1 are packed into a collection of $O(1/\delta^6)$ configurations, where each configuration is a box having a 1×1 base, that is partitioned into $O(1/\delta^2)$ (cuboidal) slots for packing big items, $O(1/\delta^6)$ slots for packing wide and long items (using NFDH), and $O(1/\delta^5)$ slots for packing small items (using 3D-NFDH). The height of a slot is the same as the height of the corresponding configuration,*
 - *The configurations are stacked one on top of the other inside the knapsack and their heights sum up to at most $1 - 2\varepsilon$,*
 - *The set T_2 contains only wide, long and small items and has a total volume of $O(\delta^4)$.*



■ **Figure 6** In Step 1, we remove at most $(1/3 + O(\varepsilon))$ fraction of the profit. In Step 2, we obtain a structured packing by [19].

Our objective is to build Area and Volume containers out of the slots for packing the wide/long and small items, respectively. The trouble is that the slot dimensions might not be large enough compared to the dimensions of the packed items, which is a necessary condition for the construction of such containers (see Section 3.1). We obtain a guessable container packing by the following lemma.

- ▶ **Lemma 11.** *By losing a profit of at most δopt_1 , and rounding the height of each configuration to the next integer multiple of μ/ε , each configuration for T_1 can be further partitioned into $O(1/\delta^6)$ guessable containers. Moreover, the items of T_2 can be packed inside a Steinberg container of height ε .*

After applying the above lemma, the rounded heights of all the configurations for T_1 sum up to at most $(1 - 2\varepsilon) + \frac{\mu}{\varepsilon} \cdot O(\frac{1}{\delta^6}) \leq 1 - \varepsilon$, for sufficiently small μ . Since items of T_2 are packed inside a Steinberg container of height ε , all the containers fit inside the knapsack and we obtain the following theorem overall.

► **Theorem 12.** *We can repack a subset of $\text{OPT} \cap I_1$ with profit at least $(\frac{2}{3} - O(\varepsilon)) \text{opt}_1$ into $O(1/\delta^{12})$ guessable Stack, Area, and Volume containers.*

An equivalent restructuring of $\text{OPT} \cap I_2$ and $\text{OPT} \cap I_3$ gives the following corollary.

► **Corollary 13.** $\text{opt}_{\text{gs}} \geq (\frac{2}{3} - O(\varepsilon)) \max\{\text{opt}_1, \text{opt}_2, \text{opt}_3\}$.

4.2 A simple $(5 + \varepsilon)$ -approximation

We now present a simple $(5 + \varepsilon)$ -approximation for 3DK, which already improves upon the $(7 + \varepsilon)$ -approximation [10]. For this, first, we provide the following packing guarantee.

► **Lemma 14.** *Let T be a set of items such that each $i \in T$ satisfies $h_i \leq \varepsilon^4$. If $v(T) \leq 1/4$, then there exists $T' \subseteq T$ with $p(T') \geq (1 - \varepsilon)p(T)$ such that there is a guessable container packing of T' into a Stack and a Steinberg container.*

Then the following lemma, together with Theorem 7, gives a $(5 + \varepsilon)$ -approximation.

► **Lemma 15.** $\text{opt} \leq (5 + \varepsilon)\text{opt}_{\text{gs}}$.

Proof. Trivially we have $\text{opt}_{\text{gs}} \geq \text{opt}_L$, since we can create a Stack container for every item in $\text{OPT} \cap L$ (at most $1/\mu^3$ such items). Hence if $\text{opt}_L > \text{opt}/5$, we are already done. Otherwise we have $\text{opt}_1 + \text{opt}_2 + \text{opt}_3 \geq \frac{4}{5}\text{opt}$. Now since $v_1 + v_2 + v_3 \leq 1$, there must exist an $i \in [3]$ such that $\frac{\text{opt}_i}{v_i} \geq \frac{4}{5}\text{opt}$ – w.l.o.g. assume $i = 1$. We have two cases.

Case 1: $v_1 \leq \frac{1}{4}$. In this case, Lemma 14 implies that $\text{opt}_{\text{gs}} \geq (1 - \varepsilon)\text{opt}_1$ (since each item in I_1 has height at most $\mu < \varepsilon^4$). Hence if $\text{opt}_1 > \text{opt}/5$, we are done. Otherwise we have $\text{opt}_2 + \text{opt}_3 \geq \frac{3}{5}\text{opt}$. W.l.o.g. assume that $\text{opt}_2 \geq \frac{3}{10}\text{opt}$, Corollary 13 gives us $\text{opt}_{\text{gs}} \geq (\frac{2}{3} - O(\varepsilon)) \cdot \frac{3}{10}\text{opt} = (\frac{1}{5} - O(\varepsilon)) \text{opt}$.

Case 2: $v_1 > \frac{1}{4}$. In this case, we sort the items of $\text{OPT} \cap I_1$ in non-increasing order of their profit to volume ratio, and pick the maximal prefix T whose volume does not exceed $\frac{1}{4}$. Then we have $v(T) \geq \frac{1}{4} - \mu$. Applying Lemma 14 to T , we obtain $\text{opt}_{\text{gs}} \geq (1 - \varepsilon) \left(\frac{1/4 - \mu}{v_1}\right) \text{opt}_1 \geq (\frac{1}{5} - O(\varepsilon)) \text{opt}$, thus completing the proof. ◀

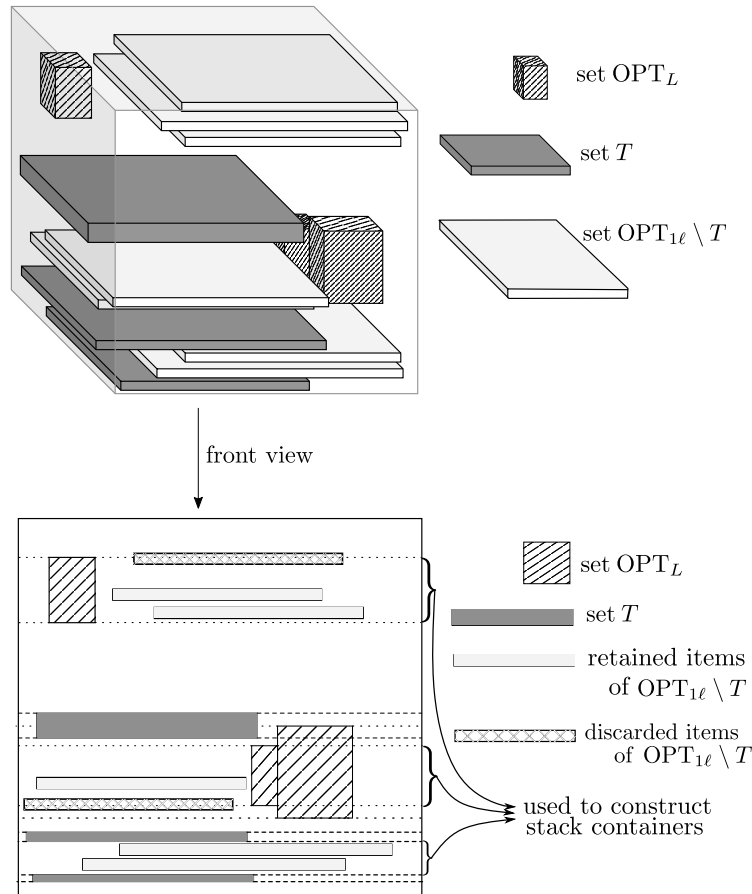
The ratio of $(5 + \varepsilon)$ is tight. One such instance is given when $v_1 = \frac{1}{8}, v_2 = v_3 = \frac{3}{8}$ and $\text{opt}_1 = \text{opt}_L = \frac{1}{5}\text{opt}$ and $\text{opt}_2 = \text{opt}_3 = \frac{3}{10}\text{opt}$.

We will obtain further improvements by using the lower bounds in the following sections.

4.3 Second lower bound on opt_{gs}

In this subsection, we consider the packing of $\text{OPT} \cap (I_{1\ell} \cup L)$, and restructure it to obtain a guessable container packing, by losing only an $O(\varepsilon)$ -fraction of the profit. Let $\text{OPT}_{1\ell} := \text{OPT} \cap I_{1\ell}$ and $\text{OPT}_L := \text{OPT} \cap L$. Since each item in L has all side lengths at least μ , the number of such items in the packing is $|\text{OPT}_L| \leq 1/\mu^3$. Also, since the width and depth of each item in $I_{1\ell}$ is at least $1/2$, the items of $\text{OPT}_{1\ell}$ must be packed in layers, i.e., the top and bottom faces of any item in $\text{OPT}_{1\ell}$ when extended parallel to the bottom face of the knapsack will not intersect another item from $\text{OPT}_{1\ell}$. We now discretize the positions of the items in the packing.

► **Lemma 16.** *The distance of the left (resp. front) face of any item from the left (resp. front) face of the knapsack can be assumed to lie in a polynomial-sized set.*



■ **Figure 7** Packing of items in $OPT \cap (L \cup I_{1\ell})$: The dotted (resp. dashed) lines show (front view of) horizontal planes passing through the top and bottom faces of items in OPT_L (resp. T). We discard the items in $OPT_{1\ell} \setminus T$ that are cut by these planes.

Let $T \subseteq OPT_{1\ell}$ be the k items of largest profit in $OPT_{1\ell}$, for some k to be chosen later. Our goal is to obtain a container packing of all items in OPT_L , together with a subset of $OPT_{1\ell}$ having a profit of at least $(1 - \varepsilon)\text{opt}_{1\ell}$. To this end, we shall pack all items in T and ensure that the number of discarded items of $OPT_{1\ell}$ is bounded by ck , for some constant c . By a result of [18], the profit of the discarded items can be bounded by $\varepsilon\text{opt}_{1\ell}$, for appropriately chosen k . We now outline the details of our restructuring procedure. See Figure 7.

We draw horizontal planes passing through the top and bottom faces of the items in $T \cup OPT_L$, and discard the items of $OPT_{1\ell} \setminus T$ intersecting these planes (see Figure 7). The number of discarded items is bounded by $2|OPT_L| = O(1/\mu^3)$. The remaining items of $OPT_{1\ell} \setminus T$ are now completely packed inside at most $k + 2|OPT_L| + 1$ horizontal strips demarcated by the planes. Each such strip is a cuboidal box that is pierced from top to bottom by at most $1/\mu^2$ Large items. Further, there are only polynomially-many positions of these Large items inside the strip owing to Lemma 16. We focus on one such strip \mathcal{S} .

► **Lemma 17.** *The items of $OPT_{1\ell} \setminus T$ present inside the strip \mathcal{S} can be repacked into at most $O(1/\mu^8)$ Stack containers, whose widths and depths lie in a polynomial-sized set.*

For each item in $T \cup OPT_L$, we create a Stack container containing the single item itself. Together with Lemma 17, we get $O_\mu(1)$ number of Stack containers inside the knapsack. We now discretize the heights of the Stack containers, so that they lie in a polynomial-sized set.

► **Lemma 18.** *By discarding $O(1/\mu^{11})k$ items from $OPT_{1\ell} \setminus T$, the height of each Stack container can be made of the form $j\bar{h}$, where $j \in [n]$ and \bar{h} is the height of some item in $I_{1\ell} \cup L$.*

Together with the $O(1/\mu^3)$ items that were discarded previously, Lemma 18 gives us that the total number of discarded items so far is at most $O(1/\mu^{11})k$. We now use the following lemma from [18].

► **Lemma 19** ([18]). *Let $p_1 \geq p_2 \geq \dots \geq p_n > 0$ be a sequence of real numbers and $P = \sum_{i=1}^n p_i$. Let c be a nonnegative integer and $\varepsilon > 0$. If $n = c^{\Omega(1/\varepsilon)}$, then there is an integer $k \leq c^{O(1/\varepsilon)}$ such that $p_{k+1} + \dots + p_{k+c} \leq \varepsilon P$.*

The above lemma invoked with $c = O(1/\mu^{11})$ and with the correct guess of k gives us that the total profit of the discarded items is at most $\varepsilon \text{opt}_{1\ell}$, assuming that $|OPT_{1\ell}|$ was sufficiently large (at least $n_0 := (1/\mu)^{\Omega(1/\varepsilon)}$) to begin with (otherwise, it is trivial). The following theorem summarizes the arguments of this section.

► **Theorem 20.** *Consider the packing of $OPT \cap (I_{1\ell} \cup L)$. Then by discarding items of $OPT \cap I_{1\ell}$ having a total profit of at most $\varepsilon \text{opt}_{1\ell}$, the remaining items can be repacked into a guessable container packing, wherein items of L are placed inside at most $1/\mu^3$ Stack containers, and items of $I_{1\ell}$ are packed into at most $(1/\mu)^{O(1/\varepsilon)}$ Stack containers.*

4.4 Third lower bound on opt_{gs}

In this subsection, we restructure the packing of items in $OPT \cap I_1$. Recall that $\text{opt}_{1\ell} = p(OPT \cap I_{1\ell})$ and $\text{opt}_{1s} = p(OPT \cap I_{1s})$, so that $\text{opt}_{1\ell} + \text{opt}_{1s} = \text{opt}_1$. Also, $v_1 := v(OPT \cap I_1)$ and $v_{1s} = v(OPT \cap I_{1s})$.

► **Theorem 21.** *Consider the packing of $OPT \cap I_1$. It is possible to repack subsets of $OPT \cap I_{1\ell}$ and $OPT \cap I_{1s}$ into a guessable Stack and Steinberg container, respectively, whose profit, say P , satisfies the following properties:*

- $P \geq (1 - O(\varepsilon)) \frac{\text{opt}_{1s}}{\max\{3v_{1s}, 1\}}$,
- If $v_1 \leq 1/3$, then $P \geq (1 - O(\varepsilon)) \left(\frac{3}{4} \text{opt}_{1\ell} + \text{opt}_{1s} \right)$,
- If $v_1 \leq 1/4$, then $P \geq (1 - O(\varepsilon)) \text{opt}_1$.

4.5 Fourth lower bound on opt_{gs}

In the previous sections, we repacked items from exactly one set out of I_1, I_2, I_3 . In this subsection, we restructure the packing of a subset of items from I_1 , together with items from $I_2 \cup I_3 \cup L$. To this end, we introduce a classification of the items in $I_2 \cup I_3 \cup L$. Let S_2 (resp. S_3, S) be the set of items in I_2 (resp. I_3, L) with height at most $1/2$. Define $\text{opt}_{2t} := p(OPT \cap S_2)$ and $\text{opt}_{2h} := p(OPT \cap (I_2 \setminus S_2))$, so that $\text{opt}_{2t} + \text{opt}_{2h} = \text{opt}_2$. The quantities opt_{3t} and opt_{3h} are defined analogously. Let $\text{opt}_{Lt} := p(OPT \cap S)$ and $\text{opt}_{Lh} := p(OPT \cap (L \setminus S))$ so that $\text{opt}_{Lt} + \text{opt}_{Lh} = \text{opt}_L$. The following theorem repacks items from $I_1 \cup S_2 \cup S_3 \cup S$.

► **Theorem 22.** *Consider the packing of $OPT \cap (I_1 \cup S_2 \cup S_3 \cup S)$ and assume that $v_1 \leq 1/6$ and $v_2, v_3 > 1/4$. Then there is a guessable container packing of a subset T of these items into $O(1/\mu^3)$ Stack and Steinberg containers having a profit of at least*

$$(1 - O(\varepsilon)) \left(\frac{3}{4} \text{opt}_{1\ell} + \text{opt}_{1s} + \max \left\{ \frac{3}{20} (\text{opt}_{2t} + \text{opt}_{3t}), \frac{1}{3} \text{opt}_{Lt} \right\} \right).$$

4.6 Fifth lower bound on opt_{gs}

In the previous subsection, we repacked items from S_2, S_3 or S together with items in I_1 . Now we consider items in $(I_2 \setminus S_2) \cup (I_3 \setminus S_3) \cup (L \setminus S)$ in OPT . As all these items have heights of more than $1/2$, they can not be stacked on top of each other. So, we can ignore their heights, and effectively obtain a 2D Knapsack instance. Now we use a container-based algorithm for 2DK from [22] to obtain a container packing of these items.

► **Theorem 23.** *Consider the packing of $OPT \cap ((I_2 \cup I_3 \cup L) \setminus (S_2 \cup S_3 \cup S))$. Then there is a guessable container packing into $O_\mu(1)$ Stack and Area containers each of height 1, having profit at least $(\frac{1}{2} - O(\varepsilon)) (\text{opt}_{2h} + \text{opt}_{3h} + \text{opt}_{Lh})$.*

4.7 Bounding opt_{gs}

Equipped with all the lower bounds of the preceding subsections, we establish Lemma 8 by an intricate case analysis depending on the volumes of various sets of items in the optimal packing. Since v_1, v_2, v_3 sum up to at most the volume of the knapsack which is 1, at least one among them must not exceed $1/3$. If at most one of them exceeds $1/3$, we show a stronger bound of $(\frac{9}{2} + \varepsilon)$ on the approximation ratio. Otherwise, we assume w.l.o.g. that $v_1 \leq 1/3$ and $v_2, v_3 > 1/3$. We further divide into subcases depending on the values of v_{2s} and v_{3s} . It turns out that the worst bound on opt_{gs} occurs when v_{2s}, v_{3s} both exceed $1/3$ and $v_1 \leq 1/6$. However for this case, Theorem 22 and Theorem 23 give us highly profitable packings, and we achieve an approximation ratio of $(\frac{139}{29} + \varepsilon)$. The details of the analysis are deferred to the full version [16].

5 Other variants

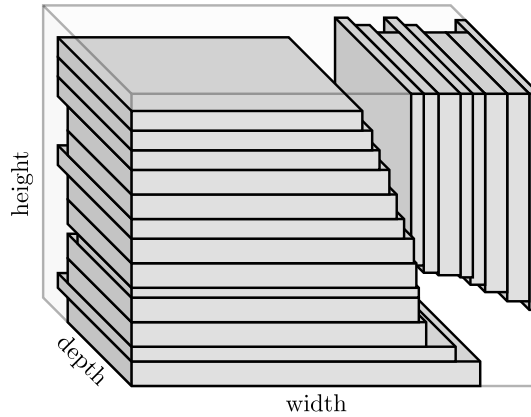
Using our techniques, we obtain further improvements for other variants: (1) cardinality case (when all items have the same profit), and (2) uniform profit-density case (when the profit of an item is equal to its volume).

► **Theorem 24.** *For any constant $\varepsilon > 0$, there is a polynomial-time $(17/4 + \varepsilon)$ -approximation algorithm for 3DK when all items have the same profit.*

► **Theorem 25.** *For any constant $\varepsilon > 0$, there is a polynomial-time $(4 + \varepsilon)$ -approximation algorithm for 3DK when the profit of an item is equal to its volume.*

5.1 Rotational case

When the items can be rotated by 90 degrees around any axis, we show the existence of improved container packing. Specifically, we introduce a new type of container called L-Container and obtain improved guarantees based on the packed volume.



■ **Figure 8** An example packing of items inside an L-Container.

L-Container. We describe L-Containers which ignore depth while packing the items (see Figure 8; two other types of L-Containers exist which ignore width, height, respectively). We first require that C must satisfy the condition $w_C \geq h_C$. The capacity of C , $\text{cap}(C)$, is given by $w_C h_C - w_C^2/4$. For each item i packed inside C , there must be some orientation with width at least $w_C/2$, depth at least $d_C/2$, and height at most εh_C . If these conditions are satisfied, then we set $f_C(i) := w_i h_i$ and ∞ otherwise. Algorithm \mathcal{A}_C is as follows. Given a set of items T satisfying $f_C(T) \leq \text{cap}(C)$, we arrange them in non-increasing order of profit/(front area) ratio. Then, we select the largest prefix T' whose total front area does not exceed $w_C h_C - w_C^2/4 - 3\varepsilon h_C^2$. We now sort the items in T' in non-increasing order of widths, start from the bottom left corner, and keep stacking them one upon each other to the maximum extent possible. Then, we rotate the remaining items by 90 degrees around the depth axis, start from the top right corner and keep packing them side by side.

The main ingredient to obtain improved approximation ratios in the rotational case is our algorithm 3DR-Vol-Pack (see Section 2.1). Although, as stated the algorithm does not pack into (guessable) containers, it can be modified to return a guessable container packing into Stack, Steinberg, and L-Containers. Using this, we obtain the following theorems for the general case, cardinality case, and uniform profit-density case, respectively.

► **Theorem 26.** *For any constant $\varepsilon > 0$, there is a polynomial-time $(30/7 + \varepsilon)$ -approximation algorithm for 3DKR.*

► **Theorem 27.** *For any constant $\varepsilon > 0$, there is a polynomial-time $(24/7 + \varepsilon)$ -approximation algorithm for 3DKR when all items have the same profit.*

► **Theorem 28.** *For any constant $\varepsilon > 0$, there is a polynomial-time $(3 + \varepsilon)$ -approximation algorithm for 3DKR when the profit of an item is equal to its volume.*

6 Conclusion

In this work, our primary focus has been to obtain improved polynomial-time approximation guarantees. Our current runtime of $n^{f(1/\varepsilon)}$ is mainly because of the step where we guess the containers and assign the items. Once this is done, every \mathcal{A}_C runs in time $O(n \log^2 n)$. Hence, one interesting direction is whether the time to guess the containers and assign the items can be brought down to $(n \cdot (\log n)^{f(1/\varepsilon)})$ using the recent techniques of [4]. Another interesting open problem is to provide a tight approximation for algorithms based on container packing.

References

- 1 Sara Ali, António Galvão Ramos, Maria Antónia Carravilla, and José Fernando Oliveira. On-line three-dimensional packing problems: A review of off-line and on-line solution approaches. *Computers & Industrial Engineering*, 168:108122, 2022. doi:10.1016/j.cie.2022.108122.
- 2 Helmut Alt. Computational aspects of packing problems. *Bulletin of EATCS*, 1(118), 2016.
- 3 Nikhil Bansal, José R. Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: inapproximability results and approximation schemes. *Mathematics of Operations Research*, 31(1):31–49, 2006. doi:10.1287/moor.1050.0168.
- 4 Moritz Buchem, Paul Deuker, and Andreas Wiese. Approximating the geometric knapsack problem in near-linear time and dynamically. In *Symposium on Computational Geometry (SoCG)*, volume 293, pages 26:1–26:14, 2024. doi:10.4230/LIPIcs.SoCG.2024.26.
- 5 Valentina Cacchiani, Manuel Iori, Alberto Locatelli, and Silvano Martello. Knapsack problems – An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems. *Computers & Operations Research*, 143:105693, 2022. doi:10.1016/j.cor.2021.105693.
- 6 Alberto Caprara. Packing d -dimensional bins in d stages. *Mathematics of Operations Research*, 33(1):203–215, 2008. doi:10.1287/moor.1070.0289.
- 7 Miroslav Chlebík and Janka Chlebíková. Hardness of approximation for orthogonal rectangle packing and covering problems. *Journal of Discrete Algorithms*, 7(3):291–305, 2009. doi:10.1016/j.jda.2009.02.002.
- 8 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. doi:10.1016/j.cosrev.2016.12.001.
- 9 Edward G. Coffman, Jr, Michael R. Garey, David S. Johnson, and Robert E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9(4):808–826, 1980. doi:10.1137/0209062.
- 10 Florian Diedrich, Rolf Harren, Klaus Jansen, Ralf Thöle, and Henning Thomas. Approximation algorithms for 3D orthogonal knapsack. *Journal of Computer Science and Technology*, 23(5):749–762, 2008. doi:10.1007/s11390-008-9170-7.
- 11 Waldo Gálvez, Fabrizio Grandoni, Salvatore Ingala, Sandy Heydrich, Arindam Khan, and Andreas Wiese. Approximating geometric knapsack via L-packings. *ACM Transactions on Algorithms*, 17(4):33:1–33:67, 2021. doi:10.1145/3473713.
- 12 Waldo Gálvez, Fabrizio Grandoni, Arindam Khan, Diego Ramírez-Romero, and Andreas Wiese. Improved approximation algorithms for 2-dimensional knapsack: Packing into multiple L-shapes, spirals, and more. In *International Symposium on Computational Geometry (SoCG)*, pages 39:1–39:17, 2021. doi:10.1007/s00453-023-01130-2.
- 13 P. C. Gilmore and Ralph E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120, 1965.
- 14 David Hillel Goodman and Ilan Garibi. *Soma Puzzle Book, The: A New Approach To The Classic Pieces*. World Scientific, 2019.
- 15 Rolf Harren. Approximation algorithms for orthogonal packing problems for hypercubes. *Theoretical Computer Science*, 410(44):4504–4532, 2009. doi:10.1016/J.TCS.2009.07.030.
- 16 Klaus Jansen, Debajyoti Kar, Arindam Khan, KVN Sreenivas, and Malte Tutas. Improved approximation algorithms for three-dimensional knapsack. *arXiv preprint arXiv:2503.19365*, 2025.
- 17 Klaus Jansen, Arindam Khan, Marvin Lira, and K. V. N. Sreenivas. A ptas for packing hypercubes into a knapsack. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 229, pages 78:1–78:20, 2022. doi:10.4230/LIPIcs.ICALP.2022.78.
- 18 Klaus Jansen and Lorant Porkolab. Improved approximation schemes for scheduling unrelated parallel machines. In *ACM Symposium on Theory of Computing (STOC)*, pages 408–417, 1999. doi:10.1145/301250.301361.

- 19 Klaus Jansen and Lars Prädél. A new asymptotic approximation algorithm for 3-dimensional strip packing. In *SOFSEM: Theory and Practice of Computer Science*, pages 327–338, 2014. doi:10.1007/978-3-319-04298-5_29.
- 20 Klaus Jansen and Guochuan Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47(3):323–342, 2007. doi:10.1007/S00453-006-0194-5.
- 21 Arindam Khan, Arnab Maiti, Amatya Sharma, and Andreas Wiese. On guillotine separable packings for the two-dimensional geometric knapsack problem. In *International Symposium on Computational Geometry (SoCG)*, volume 189, pages 48:1–48:17, 2021. doi:10.4230/LIPIcs.SoCG.2021.48.
- 22 Arindam Khan, Eklavya Sharma, and K. V. N. Sreenivas. Approximation algorithms for generalized multidimensional knapsack. *arXiv preprint*, 2021. arXiv:2102.05854.
- 23 Jeffrey C. Lagarias. *The Kepler Conjecture: The Hales-Ferguson Proof*. Springer Science & Business Media, 2011.
- 24 Yiping Lu, Danny Z. Chen, and Jianzhong Cha. Packing cubes into a cube is NP-complete in the strong sense. *Journal of Combinatorial Optimization*, 29(1):197–215, 2015. doi:10.1007/s10878-013-9701-1.
- 25 Everton Fernandes Silva, Túlio Angelo Machado Toffolo, and Tony Wauters. Exact methods for three-dimensional cutting and packing: A comparative study concerning single container problems. *Computers & Operations Research*, 109:12–27, 2019. doi:10.1016/j.cor.2019.04.020.
- 26 A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing*, 26(2):401–409, 1997. doi:10.1137/S0097539793255801.