Approximating Geometric Knapsack via L-packings

Waldo Gálvez*, Fabrizio Grandoni*, Sandy Heydrich†,
Salvatore Ingala*, Arindam Khan* and Andreas Wiese‡
* IDSIA, USI-SUPSI, Lugano, Switzerland.
Email: [waldo,fabrizio,salvatore,arindam]@idsia.ch
†MPI for Informatics and Saarbrücken Graduate School of Computer Science, Germany.
Email: heydrich@mpi-inf.mpg.de
‡Department of Industrial Engineering and Center for Mathematical Modeling, Universidad de Chile, Chile.
Email: awiese@dii.uchile.cl

***Abstract*—We study the two-dimensional geometric knapsack problem ($2DK$) in which we are given a set of n axis-aligned rectangular items, each one with an associated profit, and an axis-aligned square knapsack. The goal is to find a (non-overlapping) packing of a maximum profit subset of items inside the knapsack (without rotating items). The best-known polynomial-time approximation factor for this problem (even just in the cardinality case) is $2+\varepsilon$ [Jansen and Zhang, SODA 2004]. In this paper we break the 2 approximation barrier, achieving a polynomial-time $\frac{17}{9}+\varepsilon < 1.89$ approximation, which improves to $\frac{558}{325}+\varepsilon < 1.72$ in the cardinality case.**

Essentially all prior work on $2DK$ approximation packs items inside a constant number of rectangular containers, where items inside each container are packed using a simple greedy strategy. We deviate for the first time from this setting: we show that there exists a large profit solution where items are packed inside a constant number of containers *plus* one L-shaped region at the boundary of the knapsack which contains items that are high and narrow and items that are wide and thin. The items of these two types possibly interact in a complex manner at the corner of the L.

The above structural result is not enough however: the best-known approximation ratio for the subproblem in the L-shaped region is $2+\varepsilon$ (obtained via a trivial reduction to one-dimensional knapsack by considering tall or wide items only). Indeed this is one of the simplest special settings of the problem for which this is the best known approximation factor. As a second major, and the main algorithmic contribution of this paper, we present a PTAS for this case. We believe that this will turn out to be useful in future work in geometric packing problems.

We also consider the variant of the problem *with rotations* ($2DKR$), where items can be rotated by 90 degrees. Also in this case the best-known polynomial-time approximation factor (even for the cardinality case) is $2+\varepsilon$ [Jansen and Zhang, SODA 2004]. Exploiting part of the machinery developed for $2DK$ plus a few additional ideas, we obtain a polynomial-time $3/2+\varepsilon$-approximation for $2DKR$, which improves to $4/3+\varepsilon$ in the cardinality case.

***Index Terms*—Approximation Algorithms; Two-dimensional Knapsack; Geometric Packing; Rectangle Packing;**

The authors from IDSIA are partially supported by ERC Starting Grant NEWNET 279352 and SNSF Grant APXNET 200021_159697/1. Sandy Heydrich is in part supported by the Google Europe PhD Fellowship.

I. Introduction

The *(two-dimensional) geometric knapsack* problem ($2DK$) is the geometric variant of the classical (one-dimensional) knapsack problem. We are given a set of n items $I = \{1, \ldots, n\}$, where each item $i \in I$ is an axis-aligned open rectangle $(0, w(i)) \times (0, h(i))$ in the two-dimensional plane, and has an associated profit $p(i)$. Furthermore, we are given an axis-aligned square knapsack $K = [0, N] \times [0, N]$. W.l.o.g. we next assume that all values $w(i)$, $h(i)$, $p(i)$ and N are positive integers. Our goal is to select a subset of items $OPT \subseteq I$ of maximum total profit $opt = p(OPT) := \sum_{i \in OPT} p(i)$ and to place them so that the selected rectangles are pairwise disjoint and fully contained in the knapsack. More formally, for each $i \in OPT$ we have to define a pair of coordinates $(left(i), bottom(i))$ that specify the position of the bottom-left corner of i in the packing. In other words, i is mapped into a rectangle $R(i) := (left(i), right(i)) \times (bottom(i), top(i))$, with $right(i) = left(i)+w(i)$ and $top(i) = bottom(i)+h(i)$. For any two $i, j \in OPT$, we must have $R(i) \subseteq K$ and $R(i) \cap R(j) = \emptyset$.

Besides being a natural mathematical problem, $2DK$ is well-motivated by practical applications. For instance, one might want to place advertisements on a board or a website, or cut rectangular pieces from a sheet of some material. Also, it models a scheduling setting where each rectangle corresponds to a job that needs some "consecutive amount" of a given resource (memory storage, frequencies, etc.). In all these cases, dealing with rectangular shapes only is a reasonable simplification and often the developed techniques can be extended to deal with more general instances.

$2DK$ is NP-hard [1], and it was intensively studied from the point of view of approximation algorithms. The best known polynomial time approximation algorithm for it is due to Jansen and Zhang and yields a $(2+\varepsilon)$-approximation [2]. This is the best known result even in the *cardinality* case (with all profits being 1). However, there are reasons to believe that much better polynomial time approximation ratios are possible: there is a QPTAS under the assumption that $N = n^{\text{poly}(\log n)}$ [3], and there are PTASs if the profit of each item equals its area [4], if the size of the knapsack can be slightly increased (resource augmentation) [5], [6], if all items are relatively small [7] and if all input items are squares [8], [9]. Note that, with no restriction on N, the current best approximation for $2DK$ is $2+\varepsilon$ *even in quasi-polynomial time*[1].

All prior polynomial-time approximation algorithms for $2DK$ implicitly or explicitly exploit a *container-based* packing approach. The idea is to partition the knapsack into a constant number of axis-aligned rectangular regions (*containers*). The sizes (and therefore positions) of these containers can be *guessed* in polynomial time. Then items are packed inside the containers in a simple way: either one next to the other from left to right or from bottom to top (similarly to the one-dimensional case), or by means of the simple greedy Next-Fit-Decreasing-Height algorithm. Indeed, also the QPTAS in [3] can be cast in this framework, with the relevant difference that the number of containers in this case is poly-logarithmic (leading to a quasi-polynomial running time).

One of the major bottlenecks to achieve approximation factors better than 2 (in polynomial-time) is that items that are high and narrow (*vertical* items) and items that are wide and thin (*horizontal* items) can interact in a very complicated way. Indeed, consider the following seemingly simple L-*packing* problem: we are given a set of items i with either $w(i) > N/2$ (horizontal items) or $h(i) > N/2$ (vertical items). Our goal is to pack a maximum profit subset of them inside an L-shaped region $L = ([0, N] \times [0, h_L]) \cup ([0, w_L] \times [0, N])$, so that horizontal (resp., vertical) items are packed in the bottom-right (resp., top-left) of L. To the best of our knowledge, the best-known approximation ratio for L-packing is $2+\varepsilon$: Remove either all vertical or all horizontal items, and then pack the remaining items by a simple reduction to one-dimensional knapsack (for which an FPTAS is known). It is unclear whether a container-based packing can achieve a better approximation factor, and we conjecture that this is not the case. As we will see, a better understanding of L-packing will play a major role in the design of improved approximation algorithms for $2DK$.

A. *Our contribution*

In this paper we break the 2-approximation barrier for $2DK$. In order to do that, we substantially deviate for the first time from *pure* container-based packings, which are, either implicitly or explicitly, at the hearth of prior work. Namely, we consider *L&C-packings* that combine $O_\varepsilon(1)$ containers *plus* one L-packing of the above type (see Fig.1.(a)), and show that one such packing has large enough profit.

While it is easy to pack almost optimally items into containers, the mentioned $2+\varepsilon$ approximation for L-packings is not sufficient to achieve altogether a better than 2 approximation factor: indeed, the items of the L-packing might carry all the profit! The main algorithmic contribution of this paper is a PTAS for the L-packing problem. It is easy to solve this problem optimally in pseudo-polynomial time $(Nn)^{O(1)}$ by means of dynamic programming. We show that a $1+\varepsilon$ approximation can be obtained by restricting the top (resp., right) coordinates of horizontal (resp., vertical) items to a proper set that can be computed in polynomial time $n^{O_\varepsilon(1)}$. Given that, one can adapt the above dynamic program to run in polynomial time.

Theorem 1. *There is a PTAS for the L-packing problem.*

In order to illustrate the power of our approach, we next sketch a simple $\frac{16}{9} + O(\varepsilon)$ approximation for the cardinality case of $2DK$ (details in Section III). By standard arguments[2] it is possible to discard *large* items with both sides longer than $\varepsilon \cdot N$. The remaining items have height or width smaller than $\varepsilon \cdot N$ (*horizontal* and *vertical* items, resp.). Let us delete all items intersecting a random vertical or horizontal strip of width $\varepsilon \cdot N$ inside the knapsack. We can pack the remaining items into $O_\varepsilon(1)$ containers by exploiting the PTAS under one-

[1]The role of N in the running time is delicate, as shown by recent results on the related *strip packing* problem [10], [11], [12], [13], [14].

[2]There can be at most $O_\varepsilon(1)$ such items in any feasible solution, and if the optimum solution contains only $O_\varepsilon(1)$ items we can solve the problem optimally by brute force.

dimensional resource augmentation for $2DK$ in [6][3]. A vertical strip deletes vertical items with $O(\varepsilon)$ probability, and horizontal ones with probability roughly proportional to their width, and symmetrically for a horizontal strip. In particular, let us call *long* the items with longer side larger than $N/2$, and *short* the remaining items. Then the above argument gives in expectation roughly one half of the profit opt_{long} of long items, and three quarters of the profit opt_{short} of short ones. This is already good enough unless opt_{long} is large compared to opt_{short}.

At this point L-packings and our PTAS come into play. We shift long items such that they form 4 stacks at the sides of the knapsack in a *ring-shaped* region, see Fig.1.(b)-(c): this is possible since any vertical long item cannot have a horizontal long item *both* at its left and at its right, and vice versa. Next we delete the least profitable of these stacks and rearrange the remaining long items into an L-packing, see Fig.1.(d). Thus using our PTAS for L-packings, we can compute a solution of profit roughly three quarters of opt_{long}. The reader might check that the combination of these two algorithms gives the claimed approximation factor.

Above we used either $O_\varepsilon(1)$ containers or one L-packing: by combining the two approaches together and with a more sophisticated case analysis we achieve the following result:

Theorem 2. *There is a polynomial-time* $\frac{558}{325} + \varepsilon < 1.72$ *approximation algorithm for cardinality* $2DK$.

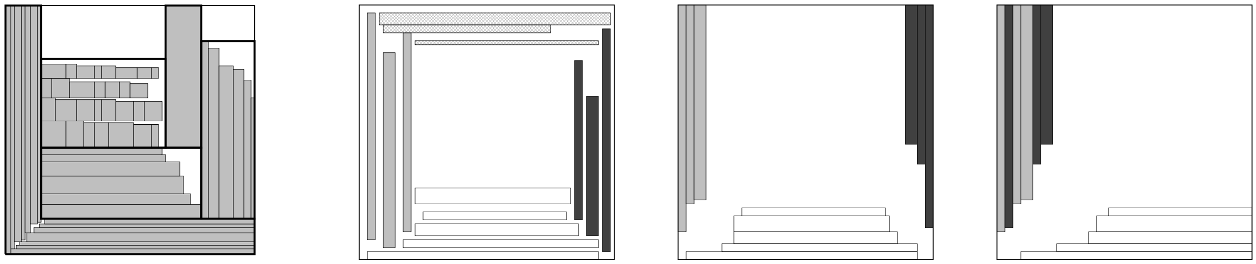
For weighted $2DK$ we face severe technical complications for proving that there is a profitable L&C-packing. One key reason is that in the weighted case we cannot discard large items since even one such item might contribute a large fraction to the optimal profit. In order to circumvent these difficulties, we exploit the *corridor-partition* at the hearth of the QPTAS for $2DK$ in [3] (in turn inspired by prior work in [15]). Roughly speaking, there exists a partition of the knapsack into $O_\varepsilon(1)$ *corridors*, consisting of the *concatenation* of $O_\varepsilon(1)$ partially overlapping rectangular regions (*subcorridors*). In [3] the authors partition the corridors into a *poly-logarithmic* number of containers. Their main algorithm then guesses these containers in time $n^{\mathrm{poly}(\log n)}$. However, we can only handle a *constant* number of containers in polynomial time. Therefore, we present a different way to partition the corridors into containers: here we lose the profit of a set of *thin* items, which in some sense play the role of long items in the previous discussion. These thin items fit in a *very narrow* ring at the boundary of the knapsack and we map them to an L-packing in the same way as in the cardinality case above. Some of the remaining non-thin items are then packed into $O_\varepsilon(1)$ containers that are placed in the (large) part of the knapsack not occupied by the L-packing. Our partition of the corridors is based on a somewhat intricate case analysis that exploits the fact that *long* consecutive subcorridors are arranged in the shape of *rings* or *spirals*: this is used to show the existence of a profitable L&C-packing.

Theorem 3. *There is a polynomial-time* $\frac{17}{9} + \varepsilon < 1.89$ *approximation algorithm for (weighted)* $2DK$.

1) Rotation setting: In the variant of $2DK$ *with rotations* ($2DKR$), we are allowed to rotate any rectangle i by 90 degrees. This means that i can also be placed in the knapsack as a rectangle of the form $(left(i), left(i)+h(i)) \times (bottom(i), bottom(i)+w(i))$. The best known polynomial time approximation factor for $2DKR$ (even for the cardinality case) is again $2+\varepsilon$ due to [2] and the mentioned QPTAS in [3] works also for this case.

By using the techniques described above and exploiting a few more ideas, we are also able to improve the approximation factor for $2DKR$. The basic idea is that any thin item can now be packed inside a narrow vertical strip (say at the right edge of the knapsack) by possibly rotating it. This way we do not lose one quarter of the profit due to the mapping to an L-packing and instead place all items from the ring into the mentioned strip (while we ensure that their total width is small). The remaining short items are packed by means of a novel *resource contraction* lemma: unless there is one *huge item* that occupies almost the whole knapsack (a case that we consider separately), we can pack almost one half of the profit of non-thin items in a *reduced* knapsack where one of the two sides is shortened by a factor $1-\varepsilon$ (hence leaving enough space for the vertical strip). We remark that here we heavily exploit the possibility to rotate items. Thus, roughly speaking, we obtain either all profit of non-thin items, or all profit of thin items plus one half of the profit of non-thin items: this gives a $3/2+\varepsilon$ approximation. A further refinement of this approach yields a $4/3+\varepsilon$ approximation in the cardinality case. We remark that, while resource augmentation is a well-established notion in approximation algorithms, resource contraction seems to be a rather novel direction to

[3]Technically this PTAS is not container-based, however we can show that it can be cast in that framework. Our version of the PTAS simplifies the algorithms and works also in the case with rotations: this might be a handy black-box tool.



(a) (b) (c) (d)

Figure 1: (a) An L&C-packing with 4 containers, where the top-left container is packed by means of Next-Fit-Decreasing-Height. (b) A subset of long items. (c) Such items are shifted into 4 stacks at the sides of the knapsack, and the top stack is deleted. (d) The final packing into an L-shaped region.

explore.

Theorem 4. *For any constant $\varepsilon > 0$, there exists a polynomial-time $\frac{3}{2} + \varepsilon$ approximation algorithm for $2DKR$. In the cardinality case the approximation factor can be improved to $\frac{4}{3} + \varepsilon$.*

B. Other related work

The mentioned $(2 + \varepsilon)$-approximation for two-dimensional knapsack [2] works in the weighted case of the problem. However, in the unweighted case a simpler $(2 + \varepsilon)$-approximation is known [16]. If one can increase the size of the knapsack by a factor $1 + \varepsilon$ in both dimensions then one can compute a solution of optimal weight, rather than an approximation, in time $f(1/\varepsilon) \cdot n^{O(1)}$ where the exponent of n does not depend on ε [9] (for some suitable function f). Similarly, for the case of squares there is a $(1+\varepsilon)$-approximation algorithm known with such a running time, i.e., an EPTAS [9]. This improves previous results such as a $(5/4 + \varepsilon)$-approximation [17] and the mentioned PTAS [8]. Two-dimensional knapsack is the separation problem when we want to solve the configuration-LP for two-dimensional bin-packing. Even though we do not have a PTAS for the former problem, Bansal et al. [4] show how to solve the latter LP to an $(1 + \varepsilon)$-accuracy using their PTAS for two-dimensional knapsack for the special case where the profit of each item equals its area. The best known (asymptotic) result for two-dimensional bin packing is due to Bansal and Khan [18] and it is based on this configuration-LP, achieving an approximation ratio of 1.405 [19] which improves a series of previous results [6], [20], [21], [22], [23]. See also the recent survey in [24].

II. A PTAS FOR L-PACKINGS

In this section we present a PTAS for the problem of finding an optimal L-packing. In this problem we are given a set of *horizontal* items I_{hor} with width larger than $N/2$, and a set of *vertical* items I_{ver} with height larger than $N/2$. Furthermore, we are given an L-shaped region $L = ([0, N] \times [0, h_L]) \cup ([0, w_L] \times [0, N])$. Our goal is to pack a subset $OPT \subseteq I := I_{hor} \cup I_{ver}$ of maximum total profit $opt = p(OPT) := \sum_{i \in OPT} p(i)$, such that $OPT_{hor} := OPT \cap I_{hor}$ is packed inside the *horizontal box* $[0, N] \times [0, h_L]$ and $OPT_{ver} := OPT \cap I_{ver}$ is packed inside the *vertical box* $[0, w_L] \times [0, N]$. We remark that packing horizontal and vertical items independently is not possible due to the possible overlaps in the intersection of the two boxes: this is what makes this problem non-trivial, in particular harder than standard (one-dimensional) knapsack.

Observe that in an optimal packing we can assume w.l.o.g. that items in OPT_{hor} are pushed as far to the right/bottom as possible. Furthermore, the items in OPT_{hor} are packed from bottom to top in non-increasing order of width. Indeed, it is possible to *permute* any two items violating this property while keeping the packing feasible. A symmetric claim holds for OPT_{ver}. See Fig. 1.(d) for an illustration.

Given the above structure, it is relatively easy to define a dynamic program (DP) that computes an optimal L-packing in pseudo-polynomial time $(Nn)^{O(1)}$. The basic idea is to scan items of I_{hor} (resp. I_{ver}) in decreasing order of width (resp., height), and each time *guess* if they are part of the optimal solution OPT. At each step either both the considered horizontal item i and vertical item j are not part of the optimal solution, or there exist a *guillotine cut*[4] separating i or j from the rest of OPT. Depending on the cases, one can define a smaller L-packing sub-instance (among N^2 choices) for which the DP table already contains a solution.

In order to achieve a $(1+\varepsilon)$-approximation in polynomial time $n^{O_\varepsilon(1)}$, we show that it is possible (with a small loss in the profit) to restrict the possible top coordinates of OPT_{hor} and right coordinates of OPT_{ver} to proper polynomial-size subsets $\mathcal{T}$ and $\mathcal{R}$, resp. We call such an L-packing $(\mathcal{T},\mathcal{R})$-*restricted*. By adapting the above DP one obtains:

Lemma 5. *An optimal $(\mathcal{T},\mathcal{R})$-restricted L-packing can be computed in time polynomial in $m := n + |\mathcal{T}| + |\mathcal{R}|$ using dynamic programming.*

We will show that there exists a $(\mathcal{T},\mathcal{R})$-restricted L-packing with the desired properties.

Lemma 6. *There exists a $(\mathcal{T},\mathcal{R})$-restricted L-packing solution of profit at least $(1-2\varepsilon)opt$, where the sets $\mathcal{T}$ and $\mathcal{R}$ have cardinality at most $n^{O(1/\varepsilon^{1/\varepsilon})}$ and can be computed in polynomial time based on the input (without knowing OPT).*

Lemmas 5 and 6 together immediately imply a PTAS for L-packings (showing Theorem 1). The rest of this section is devoted to the proof of Lemma 6.

We will describe a way to delete a subset of items $D_{hor} \subseteq OPT_{hor}$ with $p(D_{hor}) \leq 2\varepsilon p(OPT_{hor})$, and *shift down* the remaining items $OPT_{hor} \setminus D_{hor}$ so that their top coordinate belongs to a set $\mathcal{T}$ with the desired properties. Symmetrically, we will delete a subset of items $D_{ver} \subseteq OPT_{ver}$ with $p(D_{ver}) \leq 2\varepsilon p(OPT_{ver})$, and *shift to the left* the remaining items $OPT_{ver} \setminus D_{ver}$ so that their right coordinate belongs to a set $\mathcal{R}$ with the desired properties. We remark that shifting down (resp. to the left) items of OPT_{hor} (resp., OPT_{ver}) cannot create any overlap with items of OPT_{ver} (resp., OPT_{hor}). This allows us to reason on each such set separately.

We next focus on OPT_{hor} only: the construction for OPT_{ver} is symmetric. For notational convenience we let $1,\ldots,n_{hor}$ be the items of OPT_{hor} in non-increasing order of width *and* from bottom to top in the starting optimal packing. We remark that this sequence is not necessarily sorted (increasingly or decreasingly) in terms of item heights: this makes our construction much more complicated.

[4]A guillotine cut is an infinite, axis-parallel line ℓ that partitions the items in a given packing in two subsets without intersecting any item.

Let us first introduce some useful notation. Consider any subsequence $B = \{b_{start},\ldots,b_{end}\}$ of consecutive items (*interval*). For any $i \in B$, we define $top_B(i) := \sum_{k\in B, k\leq i} h(k)$ and $bottom_B(i) = top_B(i) - h(i)$. The *growing subsequence* $G = G(B) = \{g_1,\ldots,g_h\}$ of B (with possibly non-contiguous items) is defined as follows. We initially set $g_1 = b_{start}$. Given the item g_i, g_{i+1} is the smallest-index (i.e., lowest) item in $\{g_i+1,\ldots,b_{end}\}$ such that $h(g_{i+1}) \geq h(g_i)$. We halt the construction of G when we cannot find a proper g_{i+1}. For notational convenience, define $g_{h+1} = b_{end}+1$. We let $B_i^G := \{g_i+1,\ldots,g_{i+1}-1\}$ for $i = 1,\ldots,h$. Observe that the sets B_i^G partition $B \setminus G$. We will crucially exploit the following simple property.

Lemma 7. *For any $g_i \in G$ and any $j \in \{b_{start},\ldots,g_{i+1}-1\}$, $h(j) \leq h(g_i)$.*

Proof. The items $j \in B_i^G = \{g_i+1,\ldots,g_{i+1}-1\}$ have $h(j) < h(g_i)$. Indeed, any such j with $h(j) \geq h(g_i)$ would have been added to G, a contradiction.

Consider next any $j \in \{b_{start},\ldots g_i-1\}$. If $j \in G$ the claim is trivially true by construction of G. Otherwise, one has $j \in B_k^G$ for some $g_k \in G$, $g_k < g_i$. Hence, by the previous argument and by construction of G, $h(j) < h(g_k) \leq h(g_i)$. □

The intuition behind our construction is as follows. Consider the growing sequence $G = G(OPT_{hor})$, and suppose that $p(G) \leq \varepsilon \cdot p(OPT_{hor})$. Then we might simply delete G, and shift the remaining items $OPT_{hor} \setminus G = \cup_j B_j^G$ as follows. Let $\lceil x \rceil_y$ denote the smallest multiple of y not smaller than x. We consider each set B_j^G separately. For each such set, we define a baseline vertical coordinate $base_j = \lceil bottom(g_j) \rceil_{h(g_j)/2}$, where $bottom(g_j)$ is the bottom coordinate of g_j in the original packing. We next round up the height of $i \in B_j^G$ to $\hat{h}(i) := \lceil h(i) \rceil_{h(g_j)/(2n)}$, and pack the rounded items of B_j^G as low as possible above the baseline. The reader might check that the possible top coordinates for rounded items fall in a polynomial size set (using Lemma 7). It is also not hard to check that items are *not* shifted up.

We use recursion in order to handle the case $p(G) > \varepsilon \cdot p(OPT_{hor})$. Rather than deleting G, we consider each B_j^G and build a new growing subsequence for each such set. We repeat the process recursively for r_{hor} many rounds. Let $\mathcal{G}^r$ be the union of all the growing subsequences in the recursive calls of level r. Since the sets $\mathcal{G}^r$ are disjoint by construction, there must exist a value $r_{hor} \leq \frac{1}{\varepsilon}$ such that $p(\mathcal{G}^{r_{hor}}) \leq \varepsilon \cdot p(OPT_{hor})$. Therefore we can apply the same shifting argument to all growing subsequences of level r_{hor} (in particular we delete all of

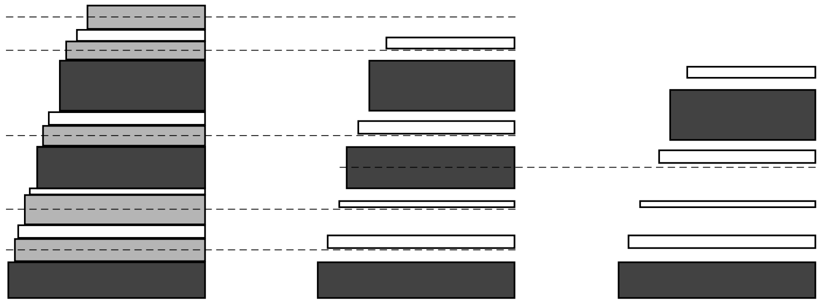


Figure 2: Illustration of the `delete&shift` procedure with $r_{hor} = 2$. The dashed lines indicate the value of the new baselines in the different stages of the algorithm. (Left) The starting packing. Dark and light grey items denote the growing sequences for the calls with $r = 2$ and $r = 1$, resp. (Middle) The shift of items at the end of the recursive calls with $r = 1$. Note that light grey items are all deleted, and dark grey items are not shifted. (Right) The shift of items at the end of the process. Here we assume that the middle dark grey item is deleted.

them). In the remaining growing subsequences we can afford to delete 1 out of $1/\varepsilon$ consecutive items (with a small loss of the profit), and then apply a similar shifting argument.

We next describe our approach in more detail. We exploit a recursive procedure `delete&shift`. This procedure takes as input two parameters: an interval $B = \{b_{start}, \ldots, b_{end}\}$, and an integer *round parameter* $r \geq 1$. Procedure `delete&shift` returns a set $D(B) \subseteq B$ of deleted items, and a shift function $shift : B \setminus D(B) \rightarrow \mathbb{N}$. Intuitively, $shift(i)$ is the value of the top coordinate of i in the shifted packing w.r.t. a proper baseline value which is implicitly defined. We initially call `delete&shift`(OPT_{hor}, r_{hor}), for a proper $r_{hor} \in \{1, \ldots, \frac{1}{\varepsilon}\}$ to be fixed later. Let $(D, shift)$ be the output of this call. The desired set of deleted items is $D_{hor} = D$, and in the final packing $top(i) = shift(i)$ for any $i \in OPT_{hor} \setminus D_{hor}$ (the right coordinate of any such i is N).

The procedure behaves differently in the cases $r = 1$ and $r > 1$. If $r = 1$, we compute the growing sequence $G = G(B) = \{g_1 = b_{start}, \ldots, g_h\}$, and set $D(B) = G(B)$. Consider any set $B_j^G = \{g_j + 1, \ldots, g_{j+1} - 1\}$, $j = 1, \ldots, h$. Let $base_j := \lceil bottom_B(g_j) \rceil_{h(g_j)/2}$. We define for any $i \in B_j^G$,

$$shift(i) = base_j + \sum_{k \in B_j^G, k \leq i} \lceil h(k) \rceil_{h(g_j)/(2n)}.$$

Observe that $shift$ is fully defined since $\cup_{j=1}^{h} B_j^G = B \setminus D(B)$.

If instead $r > 1$, we compute the growing sequence $G = G(B) = \{g_1 = b_{start}, \ldots, g_h\}$. We next delete a subset of items $D' \subseteq G$. If $h < \frac{1}{\varepsilon}$, we let $D' = D'(B) = \emptyset$. Otherwise, let $G_k = \{g_j \in G : j = k \pmod{1/\varepsilon}\} \subseteq G$, for $k \in \{0, \ldots, 1/\varepsilon - 1\}$. We set $D' = D'(B) = \{d_1, \ldots, d_p\} = G_x$ where $x = \arg\min_{k \in \{0, \ldots, 1/\varepsilon - 1\}} p(G_k)$.

Proposition 8. *One has $p(D') \leq \varepsilon \cdot p(G)$. Furthermore, any subsequence $\{g_x, g_{x+1}, \ldots, g_y\}$ of G with at least $1/\varepsilon$ items contains at least one item from D'.*

Consider each set $B_j^G = \{g_j + 1, \ldots, g_{j+1} - 1\}$, $j = 1, \ldots, h$: We run `delete&shift`$(B_j^G, r - 1)$. Let $(D_j, shift_j)$ be the output of the latter procedure, and $shift_j^{max}$ be the maximum value of $shift_j$. We set the output set of deleted items to $D(B) = D' \cup (\cup_{j=1}^{h} D_j)$.

It remains to define the function $shift$. Consider any set B_j^G, and let d_q be the deleted item in D' with largest index (hence in topmost position) in $\{b_{start}, \ldots, g_j\}$: define $base_q = \lceil bottom_B(d_q) \rceil_{h(d_q)/2}$. If there is no such d_q, we let $d_q = 0$ and $base_q = 0$. For any $i \in B_j^G$ we set:

$$\begin{aligned} shift(i) &= base_q + \textstyle\sum_{g_k \in G, d_q < g_k \leq g_j} h(g_k) \\ &+ \textstyle\sum_{g_k \in G, d_q \leq g_k < g_j} shift_k^{max} + shift_j(i). \end{aligned}$$

Analogously, if $g_j \neq d_q$, we set

$$\begin{aligned} shift(g_j) &= base_q + \textstyle\sum_{g_k \in G, d_q < g_k \leq g_j} h(g_k) \\ &+ \textstyle\sum_{g_k \in G, d_q \leq g_k < g_j} shift_k^{max}. \end{aligned}$$

This concludes the description of `delete&shift`. We next show that the final packing has the desired properties. Next lemma shows that the total profit of deleted items is small for a proper choice of the starting round parameter r_{hor}.

Lemma 9. *There is a choice of $r_{hor} \in \{1, \ldots, \frac{1}{\varepsilon}\}$ such that the final set D_{hor} of deleted items satisfies*

$p(D_{hor}) \leq 2\varepsilon \cdot p(OPT_{hor})$.

Proof. Let $\mathcal{G}^r$ denote the union of the sets $G(B)$ computed by all the recursive calls with input round parameter r. Observe that by construction these sets are disjoint. Let also $\mathcal{D}^r$ be the union of the sets $D'(B)$ on those calls (the union of sets $D(B)$ for $r = r_{hor}$). By Proposition 8 and the disjointness of sets $\mathcal{G}^r$ one has

$$\begin{aligned} p(D_{hor}) &= \textstyle\sum_{1\leq r\leq r_{hor}} p(\mathcal{D}^r) \\ &\leq \varepsilon \cdot \textstyle\sum_{r<r_{hor}} p(\mathcal{G}^r) + p(\mathcal{D}^{r_{hor}}) \\ &\leq \varepsilon \cdot p(OPT_{hor}) + p(\mathcal{D}^{r_{hor}}). \end{aligned}$$

Again by the disjointness of sets $\mathcal{G}^r$ (hence $\mathcal{D}^r$), there must exists a value of $r_{hor} \in \{1, \ldots, \frac{1}{\varepsilon}\}$ such that $p(\mathcal{D}^{r_{hor}}) \leq \varepsilon \cdot p(OPT_{hor})$. The claim follows. □

Next lemma shows that, intuitively, items are only shifted down w.r.t. the initial packing.

Lemma 10. *Let $(D, shift)$ be the output of some execution of* `delete&shift`(B, r). *Then, for any $i \in B \setminus D$, $shift(i) \leq top_B(i)$.*

Proof. We prove the claim by induction on r. Consider first the case $r = 1$. In this case, for any $i \in B_j^G$:

$$\begin{aligned} & shift(i) \\ =& \lceil bottom_B(g_j)\rceil_{h(g_j)/2} + \sum_{k\in B_j^G, k\leq i} \lceil h(k)\rceil_{h(g_j)/(2n)} \\ \leq& top_B(g_j) - \frac{1}{2}h(g_j) + \sum_{k\in B_j^G, k\leq i} h(k) + n \cdot \frac{h(g_j)}{2n} \\ =& top_B(i). \end{aligned}$$

Assume next that the claim holds up to round parameter $r - 1 \geq 1$, and consider round r. For any $i \in B_j^G$ with $base_q = \lceil bottom_B(d_q)\rceil_{h(d_q)/2}$, one has

$$\begin{aligned} & shift(i) \\ =& \lceil bottom_B(d_q)\rceil_{h(d_q)/2} + \sum_{g_k\in G, d_q<g_k\leq g_j} h(g_k) \\ +& \sum_{g_k\in G, d_q\leq g_k<g_j} shift_k^{max} + shift_j(i) \\ \leq& top_B(d_q) + \sum_{g_k\in G, d_q<g_k\leq g_j} h(g_k) \\ +& \sum_{g_k\in G, d_q\leq g_k<g_j} top_{B_k^G}(g_{k+1} - 1) + top_{B_j^G}(i) \\ =& top_B(i). \end{aligned}$$

An analogous chain of inequalities shows that $shift(g_j) \leq top_B(g_j)$ for any $g_j \in G \setminus D'$. A similar proof works for the special case $base_q = 0$. □

It remains to show that the final set of values of $top(i) = shift(i)$ has the desired properties. This is the most delicate part of our analysis. We define a set $\mathcal{T}^r$ of candidate top coordinates recursively in r. Set $\mathcal{T}^1$ contains, for any item $j \in I_{hor}$, and any integer $1 \leq a \leq 4n^2$, the value $a \cdot \frac{h(j)}{2n}$. Set $\mathcal{T}^r$, for $r > 1$ is defined recursively w.r.t. to $\mathcal{T}^{r-1}$. For any item j, any integer $0 \leq a \leq 2n - 1$, any tuple of $b \leq 1/\varepsilon - 1$ items $j(1), \ldots, j(b)$, and any tuple of $c \leq 1/\varepsilon$ values $s(1), \ldots, s(c) \in \mathcal{T}^{r-1}$, $\mathcal{T}^r$ contains the sum $a \cdot \frac{h(j)}{2} + \sum_{k=1}^{b} h(j(k)) + \sum_{k=1}^{c} s(k)$. Note that sets $\mathcal{T}^r$ can be computed based on the input only (without knowing OPT). It is easy to show that $\mathcal{T}^r$ has polynomial size for $r = O_\varepsilon(1)$.

Lemma 11. *For any integer $r \geq 1$, $|\mathcal{T}^r| \leq (2n)^{\frac{r+2+(r-1)\varepsilon}{\varepsilon^{r-1}}}$.*

Proof. We prove the claim by induction on r. The claim is trivially true for $r = 1$ since there are n choices for item j and $4n^2$ choices for the integer a, hence altogether at most $n \cdot 4n^2 < 8n^3$ choices. For $r > 1$, the number of possible values of $\mathcal{T}^r$ is at most

$$\begin{aligned} & n \cdot 2n \cdot (\sum_{b=0}^{1/\varepsilon-1} n^b) \cdot (\sum_{c=0}^{1/\varepsilon} |\mathcal{T}^{r-1}|^c) \leq 4n^2 \cdot n^{\frac{1}{\varepsilon}-1} \cdot |\mathcal{T}^{r-1}|^{\frac{1}{\varepsilon}} \\ & \leq (2n)^{\frac{1}{\varepsilon}+1}((2n)^{\frac{r+1+(r-2)\varepsilon}{\varepsilon^{r-2}}})^{\frac{1}{\varepsilon}} \leq (2n)^{\frac{r+2+(r-1)\varepsilon}{\varepsilon^{r-1}}}. \end{aligned}$$

□

Next lemma shows that the values of $shift$ returned by `delete&shift` for round parameter r belong to $\mathcal{T}^r$, hence the final top coordinates belong to $\mathcal{T} := \mathcal{T}^{r_{hor}}$.

Lemma 12. *Let $(D, shift)$ be the output of some execution of* `delete&shift`(B, r). *Then, for any $i \in B \setminus D$, $shift(i) \in \mathcal{T}^r$.*

Proof. We prove the claim by induction on r. For the case $r = 1$, recall that for any $i \in B_j^G$ one has

$$\begin{aligned} shift(i) = & \lceil bottom_B(g_j)\rceil_{h(g_j)/2} \\ & + \sum_{k\in B_j^G, k\leq i} \lceil h(k)\rceil_{h(g_j)/(2n)}. \end{aligned}$$

By Lemma 7, $bottom_B(g_j) = \sum_{k\in B, k<g_j} h(k) \leq (n - 1) \cdot h(g_j)$. By the same lemma, $\sum_{k\in B_j^G, k\leq i} h(k) \leq (n-$

$1) \cdot h(g_j)$. It follows that

$$shift(i) \leq 2(n-1) \cdot h(g_j) + \frac{h(g_j)}{2} + (n-1) \cdot \frac{h(g_j)}{2n} \leq 4n^2 \cdot \frac{h(g_j)}{2n}.$$

Hence $shift(i) = a \cdot \frac{h(g_j)}{2n}$ for some integer $1 \leq a \leq 4n^2$, and $shift(i) \in \mathcal{T}^1$ for $j = g_j$ and for a proper choice of a.

Assume next that the claim is true up to $r-1 \geq 1$, and consider the case r. Consider any $i \in B_j^G$, and assume $0 < base_q = \lceil bottom_B(d_q) \rceil_{h(d_q)/2}$. One has:

$$shift(i) = \lceil bottom_B(d_q) \rceil_{h(d_q)/2} + \sum_{g_k \in G, d_q < g_k \leq g_j} h(g_k) + \sum_{g_k \in G, d_q \leq g_k < g_j} shift_k^{max} + shift_j(i).$$

By Lemma 7, $bottom_B(d_q) \leq (n-1)h(d_q)$, therefore $\lceil bottom_B(d_q) \rceil_{h(d_q)/2} = a \cdot \frac{h(d_q)}{2}$ for some integer $1 \leq a \leq 2(n-1)+1$. By Proposition 8, $|\{g_k \in G, d_q < g_k \leq g_j\}| \leq 1/\varepsilon - 1$. Hence $\sum_{g_k \in G, d_q < g_k \leq g_j} h(g_k)$ is a value contained in the set of sums of $b \leq 1/\varepsilon - 1$ item heights. By inductive hypothesis $shift_k^{max}, shift_j(i) \in \mathcal{T}^{r-1}$. Hence by a similar argument the value of $\sum_{g_k \in G, d_q \leq g_k < g_j} shift_k^{max} + shift_j(i)$ is contained in the set of sums of $c \leq 1/\varepsilon - 1 + 1$ values taken from $\mathcal{T}^{r-1}$. Altogether, $shift(i) \in \mathcal{T}^r$. A similar argument, without the term $shift_j(i)$, shows that $shift(g_j) \in \mathcal{T}^r$ for any $g_j \in G \setminus D'$. The proof works similarly in the case $base_q = 0$ by setting $a = 0$. The claim follows. □

Proof of Lemma 6. We apply the procedure `delete&shift` to OPT_{hor} as described before, and a symmetric procedure to OPT_{ver}. In particular the latter procedure computes a set $D_{ver} \subseteq OPT_{ver}$ of deleted items, and the remaining items are shifted to the left so that their right coordinate belongs to a set $\mathcal{R} := \mathcal{R}^{r_{ver}}$, defined analogously to the case of $\mathcal{T} := \mathcal{T}^{r_{hor}}$, for some integer $r_{ver} \in \{1, \ldots, 1/\varepsilon\}$ (possibly different from r_{hor}, though by averaging this is not critical).

It is easy to see that the profit of non-deleted items satisfies the claim by Lemma 9 and its symmetric version. Similarly, the sets $\mathcal{T}$ and $\mathcal{R}$ satisfy the claim by Lemmas 11 and 12, and their symmetric versions. Finally, w.r.t. the original packing non-deleted items in OPT_{hor} and OPT_{ver} can be only shifted to the bottom and to the left, resp., by Lemma 10 and its symmetric version. This implies that the overall packing is feasible. □

III. A Simple Improved Approximation for Cardinality $2DK$

In this section we present a simple improved approximation for the cardinality case of $2DK$. We can assume that the optimal solution $OPT \subseteq I$ satisfies that $|OPT| \geq 1/\varepsilon^3$ since otherwise we can solve the problem optimally by brute force in time $n^{O(1/\varepsilon^3)}$. Therefore, we can discard from the input all *large* items with both sides larger than $\varepsilon \cdot N$: any feasible solution can contain at most $1/\varepsilon^2$ such items, and discarding them decreases the cardinality of OPT at most by a factor $1+\varepsilon$. Let OPT denote this slightly sub-optimal solution obtained by removing large items.

We will need the following technical lemma, that holds also in the weighted case (see also Fig.1.(b)-(d)).

Lemma 13. *Let H and V be given subsets of items from some feasible solution with width and height strictly larger than $N/2$, resp. Let h_H and w_V be the total height and width of items of H and V, resp. Then there exists an L-packing of a set $APX \subseteq H \cup V$ with $p(APX) \geq \frac{3}{4}(p(H)+p(V))$ into the area $L = ([0,N] \times [0,h_H]) \cup ([0,w_V] \times [0,N])$.*

Proof. Let us consider the packing of $H \cup V$. Consider each $i \in H$ that has no $j \in V$ to its top (resp., to its bottom) and shift it up (resp. down) until it hits another $i' \in H$ or the top (resp, bottom) side of the knapsack. Note that, since $h(j) > N/2$ for any $j \in V$, one of the two cases above always applies. We iterate this process as long as possible to move any such i. We perform a symmetric process on V. At the end of the process all items in $H \cup V$ are stacked on the 4 sides of the knapsack[5].

Next we remove the least profitable of the 4 stacks: by a simple permutation argument we can guarantee that this is the top or right stack. We next discuss the case that it is the top one, the other case being symmetric. We show how to repack the remaining items in a boundary L of the desired size by permuting items in a proper order. In more detail, suppose that the items packed on the left (resp., right and bottom) have a total width of w_l (resp., total width of w_r and total height of h_b). We next show that there exists a packing into $L' = ([0,N] \times [0,h_b]) \cup ([0, w_l + w_r] \times [0,N])$. We prove the claim by induction. Suppose that we have proved it for all packings into left, right and bottom stacks with parameters w'_l, w'_r,

[5] It is possible to permute items in the left stack so that items appear from left to right in non-increasing order of height, and symmetrically for the other stacks. This is not crucial for this proof, but we implemented this permutation in Fig.1.(c).

and h' such that $h' < h_b$ or $w'_l + w'_r < w_l + w_r$ or $w'_l + w'_r = w_l + w_r$ and $w'_r < w_r$.

In the considered packing we can always find a guillotine cut ℓ, such that one side of the cut contains precisely one *lonely* item among the leftmost, rightmost and bottommost items. Let ℓ be such a cut. First assume that the lonely item j is the bottommost one. Then by induction the claim is true for the part above ℓ since the part of the packing above ℓ has parameters w_l, w_r, and $h - h(j)$. Thus, it is also true for the entire packing. A similar argument applies if the lonely item j is the leftmost one.

It remains to consider the case that the lonely item j is the rightmost one. We remove j temporarily and move *all* other items by $w(j)$ to the right. Then we insert j at the left (in the space freed by the previous shifting). By induction, the claim is true for the resulting packing since it has parameters $w_l + w(j)$, $w_r - w(j)$, and h, resp. □

For our algorithm, we consider the following three packings. The first uses an L that occupies the full knapsack, i.e., $w_L = h_L = N$. Let $OPT_{long} \subseteq OPT$ be the items in OPT with height or width strictly larger than $N/2$ and define $OPT_{short} = OPT \setminus OPT_{long}$. We apply Lemma 13 to OPT_{long} and hence obtain a packing for this L with a profit of at least $\frac{3}{4}p(OPT_{long})$. We run our PTAS for L-packings from Theorem 1 on this L, the input consisting of all items in I having one side longer than $N/2$. Hence we obtain a solution with profit at least $(\frac{3}{4} - O(\varepsilon))p(OPT_{long})$.

For the other two packings we employ the one-sided resource augmentation PTAS from [6]. We apply this algorithm to the slightly reduced knapsacks $[0, N] \times [0, N/(1+\varepsilon)]$ and $[0, N/(1+\varepsilon)] \times [0, N]$ such that in both cases it outputs a solution that fits in the full knapsack $[0, N] \times [0, N]$ and whose profit is by at most a factor $1 + O(\varepsilon)$ worse than the optimal solution for the respective reduced knapsacks. We will prove in Theorem 14 that one of these solutions yields a profit of at least $(\frac{1}{2} - O(\varepsilon))p(OPT) + (\frac{1}{4} - O(\varepsilon))p(OPT_{short})$ and hence one of our packings yields a $(\frac{16}{9} + \varepsilon)$-approximation.

Theorem 14. *There is a* $\frac{16}{9} + \varepsilon$ *approximation for the cardinality case of* $2DK$.

Proof. Let OPT be the considered optimal solution with $opt = p(OPT)$. Recall that there are no large items. Let also $OPT_{vert} \subseteq OPT$ be the (*vertical*) items with height more than $\varepsilon \cdot N$ (hence with width at most $\varepsilon \cdot N$), and $OPT_{hor} = OPT \setminus OPT_{ver}$ (*horizontal* items). Note that with this definition both sides of a horizontal item might have a length of at most $\varepsilon \cdot N$. We let $opt_{long} = p(OPT_{long})$ and $opt_{short} = p(OPT_{short})$.

As mentioned above, our L-packing PTAS achieves a profit of at least $(\frac{3}{4} - O(\varepsilon))opt_{long}$ which can be seen by applying Lemma 13 with $H = OPT_{long} \cap OPT_{hor}$ and $V = OPT_{long} \cap OPT_{ver}$. In order to show that the other two packings yield a good profit, consider a *random horizontal strip* $S = [0, N] \times [a, a+\varepsilon \cdot N]$ (fully contained in the knapsack) where $a \in [0, (1-\varepsilon)N)$ is chosen unformly at random. We remove all items of OPT intersecting S. Each item in OPT_{hor} and $OPT_{short} \cap OPT_{ver}$ is deleted with probability at most 3ε and $\frac{1}{2} + 2\varepsilon$, resp. Therefore the total profit of the remaining items is in expectation at least $(1 - 3\varepsilon)p(OPT_{hor}) + (\frac{1}{2} - 2\varepsilon)p(OPT_{short} \cap OPT_{vert})$. Observe that the resulting solution can be packed into a restricted knapsack of size $[0, N] \times [0, N/(1+\varepsilon)]$ by shifting down the items above the horizontal strip. Therefore, when we apply the resource augmentation algorithm in [6] to the knapsack $[0, N] \times [0, N/(1+\varepsilon)]$, up to a factor $1 - \varepsilon$, we will find a solution of (deterministically!) at least the same profit. In other terms, this profit is at least $(1 - 4\varepsilon)p(OPT_{hor}) + (\frac{1}{2} - \frac{5}{2}\varepsilon)p(OPT_{short} \cap OPT_{vert})$.

By a symmetric argument, we obtain a solution of profit at least $(1 - 4\varepsilon)p(OPT_{ver}) + (\frac{1}{2} - \frac{5}{2}\varepsilon)p(OPT_{short} \cap OPT_{hor})$ when we apply the algorithm in [6] to the knapsack $[0, N/(1+\varepsilon)] \times [0, N]$. Thus the best of the latter two solutions has profit at least $(\frac{1}{2} - 2\varepsilon)opt_{long} + (\frac{3}{4} - \frac{13}{4}\varepsilon)opt_{short} = (\frac{1}{2} - 2\varepsilon)opt + (\frac{1}{4} - \frac{5}{4}\varepsilon)opt_{short}$. The best of our three solutions has therefore value at least $(\frac{9}{16} - O(\varepsilon))opt$ where the worst case is achieved for roughly $opt_{long} = 3 \cdot opt_{short}$. □

In the above result we use either an L-packing or a container packing. The $\frac{558}{325} + \varepsilon$ approximation claimed in Theorem 2 is obtained by a careful combination of these two packings. In particular, we consider configurations where long items (or a subset of them) can be packed into a relatively small L, and pack part of the remaining short items in the complementary rectangular region (using container packings and Steinberg's algorithm [25]). The proof is based on a long and tedious case analysis, that we omit for reasons of space.

IV. Weighted case without rotations

As mentioned in the introduction, for the weighted case we exploit the corridor-partition in [3]. Due to reasons of space, we will give only the high level intuition and omit the technical details. We consider an almost optimal solution OPT. By standard arguments, we can assume that OPT does not contain any *small* item, with both sides much smaller than N (such items can be packed very accurately in the residual free space at the end of the process).

Recall that we are given a constant number of corridors, each one consisting of a constant number of subcorridors. We partition each subcorridor into a constant number of containers. We start the partition from a subcorridor that is either at the end of a corridor or that is the central subcorridor of 3 consecutive subcorridors arranged in an U-*shaped* manner. We partition this subcorridor into a constant number of containers of roughly the same size. It is possible to pack almost all items contained in the considered subcorridor into the containers. The remaining items would fit into an additional very thin container, however, our space does not suffice to add it to the rest of the packing. The constructed containers induce a partition of the rest of the corridor into a constant number of smaller corridors, and the process is then applied recursively until each subcorridor has been partitioned into containers. This yields a constant number of containers overall. We call the items F packed into the containers *fat*, and the remaining items T *thin*.

We say that a subcorridor is *long* if (essentially) its longer side is longer than $N/2$, and it is *short* otherwise. We denote by L the items that are contained in a long subcorridor and by S the remaining items. We define $LF = L \cap F$ and analogously SF, LT, and ST. We observe that if a corridor has several consecutive long subcorridors then those are arranged in the shape of *spirals* or *rings*. One can show that if a subcorridor is processed last (among all subcorridors of some corridor) in the above container partition then we can pack all its items into the containers and hence do not loose the profit of any of its items (i.e., there are no thin items in this subcorridor).

In the partitioning routine above we have some flexibility in the order in which we partition the subcorridors, which also results in different sets F and T. Depending on this order, a case analysis (involving 7 cases) shows that we can obtain container-based solutions roughly of profit either $p(LF) + p(SF)$, or $p(LF) + p(SF)/2 + p(LT)/2$, or $p(LF) + p(SF)/2 + p(ST)/2$. This is not yet sufficient to a achieve a better than 2 approximation: at this point our PTAS for L-packings comes into play. Thin items are either very wide and thin (*horizontal*) or very tall and narrow (*vertical*). In the above partition method we can enforce that the total height/width of horizontal/vertical thin items is an arbitrarily small fraction of N. Therefore, we can pack (roughly) at least three quarters of the profit of LT in a very thin L-shaped region at the boundary of the knapsack by a similar argument as in Section III, and then pack also ST in a slightly larger L-shaped region. The space left free by this L-packing is almost the entire knapsack. A random strip argument similar to the one in Section III shows that in the remaining space there is a packing with constantly many containers which achieves at least half of the profit of SF. Altogether we essentially get a profit at least $\frac{3}{4}p(LT) + p(ST) + \frac{1}{2}p(SF)$. One can show that the best solution among the ones provided above yields a $(17/9 + O(\varepsilon))$-approximation algorithm where the term $O(\varepsilon)$ is due to using PTASs for computing the actual packing and certain omitted technical details.

V. Improved Approximation for Cardinality $2DKR$

In this section we present a simple polynomial time $(3/2 + \varepsilon)$-approximation algorithm for $2DKR$ for the cardinality case. We next assume w.l.o.g. that ε is sufficiently small.

Consider some optimal solution OPT to $2DKR$, with an associated packing in the knapsack. We crucially exploit the following resource contraction lemma, which is our main new idea in the rotation case.

Lemma 15. *(Resource Contraction Lemma) For given positive constants $\varepsilon \leq 1/13$ and $\varepsilon_{small} < \varepsilon^{\frac{1}{2\varepsilon}+1}$, suppose that there exists a feasible packing of a set of items M, with $|M| \geq 1/\varepsilon_{small}^3$. Then it is possible to pack a subset $M' \subseteq M$ of cardinality at least $\frac{2}{3}(1 - O(\varepsilon))|M|$ into $[0, \left(1 - \varepsilon^{\frac{1}{2\varepsilon}+1}\right) N] \times [0, N]$ if rotations are allowed.*

Given the above lemma, it is not hard to achieve the desired approximation.

Theorem 16. *There is a $\frac{3}{2} + \varepsilon$ approximation for the cardinality case of $2DKR$.*

Proof. Let OPT be some optimal solution with an associated packing. If $|OPT| \leq \frac{1}{\varepsilon_{small}^3}$, then we can solve the problem optimally by brute force. Otherwise, by Lemma 15 there exists $OPT' \subseteq OPT$ of cardinality at least $\frac{2}{3}(1 - O(\varepsilon))|OPT|$ that can be packed inside $K' = [0, \left(1 - \varepsilon^{\frac{1}{2\varepsilon}+1}\right) N] \times [0, N]$. Therefore, applying the resource augmentation PTAS in [6] to K' with proper constants, one obtains a feasible packing of at least $|OPT'|$ items into the original knapsack. □

It remains to prove Lemma 15. W.l.o.g., assume $h(i) \geq w(i)$ for all items $i \in M$. Let us remove from M all items that are larger than $\varepsilon_{small}N$ in both dimensions. Let M_2 be the resulting set: observe that $|M_2| \geq (1-\varepsilon_{small})|M|$.

We next show how to remove from M_2 a set of cardinality at most $\varepsilon|M_2|$ such that the remaining items M_3 are either *very tall* or *not too tall*. The exact meaning will be given next. We use the notation $[k] = \{1, \ldots, k\}$ for a positive integer k.

Lemma 17. *Given any constant $1/2 > \varepsilon > 0$, there exists a value $i \in [\lceil 1/(2\varepsilon) \rceil]$ such that all items in M_2 having height in $((1-2\varepsilon^i)N, (1-\varepsilon^{i+1})N]$ have total cardinality at most $\varepsilon|M_2|$.*

Proof. Let K_i be the set of items in M_2 with height in $((1-2\varepsilon^i)N, (1-\varepsilon^{i+1})N]$ for $i \in [\lceil 1/(2\varepsilon) \rceil]$. An item can belong to at most two such sets as $\varepsilon < 1/2$. Thus, the smallest such set has cardinality at most $\varepsilon|M_2|$. $\square$

We remove from M_2 the elements from the set K_i of minimum cardinality guaranteed by the above lemma, and let M_3 be the resulting set. We also define $\varepsilon_s = \varepsilon^i$ for the same i. Thus, $\varepsilon_s \geq \varepsilon^{1/2\varepsilon} > \varepsilon_{small}/\varepsilon$. Note that the items in M_3 have height either at most $(1-2\varepsilon_s)N$ or above $(1-\varepsilon \cdot \varepsilon_s)N$.

For any $\delta > 0$ denote the strips of width N and height δN at the top and bottom of the knapsack by $S_{T,\delta} := [0,N] \times [(1-\delta)N, N]$ and $S_{B,\delta} := [0,N] \times [0, \delta N]$, resp. Similarly, denote the strips of height N and width δN to the left and right of the knapsack by $S_{L,\delta} := [0, \delta N] \times [0,N]$ and $S_{R,\delta} := [(1-\delta)N, N] \times [0,N]$, resp. The set of items in M_3 intersected by and fully contained in strip $S_{K,\delta}$ are denoted by $E_{K,\delta}$ and $C_{K,\delta}$, resp. Obviously $C_{K,\delta} \subseteq E_{K,\delta}$. Let $a(I)$ denote the total area of items in I, i.e., $a(I) = \sum_{i \in I} w(i) \cdot h(i)$.

Lemma 18. *Either $a(E_{L,\varepsilon_s} \cup E_{R,\varepsilon_s}) \leq \frac{(1+8\varepsilon_s)}{2}N^2$ or $a(E_{T,\varepsilon_s} \cup E_{B,\varepsilon_s}) \leq \frac{(1+8\varepsilon_s)}{2}N^2$.*

Proof. Let us define $V := E_{L,\varepsilon_s} \cup E_{R,\varepsilon_s}$ and $H := E_{T,\varepsilon_s} \cup E_{B,\varepsilon_s}$. Note that, $a(V) + a(H) = a(V \cup H) + a(V \cap H)$. Clearly $a(V \cup H) \leq N^2$ since all items fit into the knapsack. On the other hand, except possibly four items (the ones that contain at least one of the points $(\varepsilon_s N, \varepsilon_s N), ((1-\varepsilon_s)N, \varepsilon_s N), (\varepsilon_s N, (1-\varepsilon_s)N), ((1-\varepsilon_s)N, (1-\varepsilon_s)N))$ all other items in $V \cap H$ lie entirely within the four strips $S_{L,\varepsilon_s} \cup S_{R,\varepsilon_s} \cup S_{T,\varepsilon_s} \cup S_{B,\varepsilon_s}$. Thus $a(V \cap H) \leq 4\varepsilon_s N^2 + 4\varepsilon_{small}N^2 \leq 8\varepsilon_s N^2$, as $\varepsilon_{small} \leq \varepsilon_s$. We can conclude that $\min\{a(V), a(H)\} \leq \frac{a(V \cup H) + a(V \cap H)}{2} \leq \frac{(1+8\varepsilon_s)}{2}N^2$. $\square$

Now we state Steinberg's Theorem that we use in Lemma 20.

Theorem 19 (A. Steinberg [25]). *We are given a set of items I' and a knapsack $Q = [0,w] \times [0,h]$. Let $w_{max} \leq w$ and $h_{max} \leq h$ be the maximum width and maximum height among the items in I' respectively. Also we denote $x_+ := max(x, 0)$. If*

$$2a(I') \leq wh - (2w_{max} - w)_+(2h_{max} - h)_+$$

then I' can be packed into Q.

Lemma 20. *Given a constant $0 < \varepsilon_a < 1/2$ and a set of items $\tilde{M} := \{1, \ldots, k\}$ with $w(i) \leq \varepsilon_{small}N$ for all $i \in \tilde{M}$. If $a(\tilde{M}) \leq (1/2 + \varepsilon_a)N^2$, then a subset of $\tilde{M}$ of cardinality at least $(1 - 2\varepsilon_s - 2\varepsilon_a)|\tilde{M}|$ can be packed into $[0, (1-\varepsilon_s)N] \times [0, N]$.*

Proof. W.l.o.g., assume the items in $\tilde{M}$ are given in nondecreasing order according to their area. Note that $a(i) \leq \varepsilon_{small}N^2 \leq \frac{\varepsilon_s}{2}N^2$ for any $i \in \tilde{M}$. Let $S := \{1, \ldots, j\}$ be such that $\frac{(1-2\varepsilon_s)}{2}N^2 \leq \sum_{i=1}^{j} a(i) \leq \frac{(1-\varepsilon_s)}{2}N^2$ and $\sum_{i=1}^{j+1} a(i) > \frac{(1-\varepsilon_s)}{2}N^2$. Then from Theorem 19, S can be packed into $[0, (1-\varepsilon_s)N] \times [0, N]$. As we considered items in the order of nondecreasing area, $\frac{|S|}{|\tilde{M}|} \geq \frac{\left(\frac{1}{2} - \varepsilon_s\right)}{\left(\frac{1}{2} + \varepsilon_a\right)}$. Thus, $|S| \geq \left(1 - \frac{(\varepsilon_a + \varepsilon_s)}{\left(\frac{1}{2} + \varepsilon_a\right)}\right)|\tilde{M}| > (1 - 2\varepsilon_a - 2\varepsilon_s)|\tilde{M}|$. $\square$

From Lemma 18, we can assume w.l.o.g. that $a(E_{T,\varepsilon_s} \cup E_{B,\varepsilon_s}) \leq \frac{(1+8\varepsilon_s)}{2}N^2$. Let X be the set of items in M_3 that intersect both S_{T,ε_s} and S_{B,ε_s} and $Y := \{E_{T,\varepsilon_s} \cup E_{B,\varepsilon_s}\} \setminus X$. Define $Z := M_3 \setminus \{X \cup Y\}$ to be the rest of the items. Let us define $w(X) = \sum_{i \in X} w(i)$. Now there are two cases.

Case A. $w(X) \geq 12\varepsilon \cdot \varepsilon_s N$. From Lemma 17, all items in X intersect both $S_{T,\varepsilon \cdot \varepsilon_s}$ and $S_{B,\varepsilon \cdot \varepsilon_s}$. So the removal of $X \cup C_{T,\varepsilon \cdot \varepsilon_s} \cup C_{B,\varepsilon \cdot \varepsilon_s}$ creates a few empty strips of height N and total width of $w(X)$. By a simple permutation argument, all items in $Y \cup Z$ can be packed inside $[0, N - w(X)] \times [0, N]$, leaving an empty vertical strip of width $w(X)$ on the right side of the knapsack. Next we rotate $C_{T,\varepsilon \cdot \varepsilon_s}$ and $C_{B,\varepsilon \cdot \varepsilon_s}$ and pack them in two vertical strips, each of width $\varepsilon \cdot \varepsilon_s N$. Note that $w(i) \leq \varepsilon \cdot \varepsilon_s N$ for all $i \in X$. Now take items in X by nondecreasing width, till their total width is in $[w(X) - 4\varepsilon \cdot \varepsilon_s N, w(X) - 3\varepsilon \cdot \varepsilon_s N]$ and pack them into another vertical strip. The cardinality of this set is at least $\frac{(w(X) - 4\varepsilon \cdot \varepsilon_s N)}{w(X)}|X| \geq \frac{2}{3}|X|$, where the last inequality follow by the Case A assumption. Hence, at least $\frac{2}{3}|X| + |Y| + |Z| \geq \frac{2}{3}(|X| + |Y| + |Z|)$ items can be packed into $[0, (1 - \varepsilon \cdot \varepsilon_s)N] \times [0, N]$.

Case B. $w(X) < 12\varepsilon \cdot \varepsilon_s N$. Observe that $Y = (E_{T,\varepsilon_s} \setminus X) \dot\cup (E_{B,\varepsilon_s} \setminus X)$, hence $|Y| = |E_{T,\varepsilon_s} \setminus X| + |E_{B,\varepsilon_s} \setminus X|$. Assume w.l.o.g. that $|E_{B,\varepsilon_s} \setminus X| \geq |Y|/2 \geq |E_{T,\varepsilon_s} \setminus X|$. Then remove E_{T,ε_s}. We can pack X on top of $M \setminus$

E_{T,ε_s} as $12\varepsilon \cdot \varepsilon_s \leq \varepsilon_s - \varepsilon \cdot \varepsilon_s$ for $\varepsilon \leq 1/13$. This gives a packing of $|X| + |Z| + \frac{|Y|}{2}$. On the other hand, as $a(X \cup Y) = a(E_{T,\varepsilon_s} \cup E_{B,\varepsilon_s}) \leq \frac{(1+8\varepsilon_s)}{2} N^2$, from Lemma 20, it is possible to pack at least $(1 - 2\varepsilon_s - 8\varepsilon_s)|X \cup Y| \geq (1 - 10\varepsilon_s)(|X| + |Y|)$ many items into $[0, (1 - \varepsilon \cdot \varepsilon_s)N] \times [0, N]$.

Thus we can always pack a set of items of cardinality at least

$$\begin{aligned} & \max\{(1-10\varepsilon_s)(|X|+|Y|), |X|+|Z|+\tfrac{|Y|}{2}\} \\ \geq\ & \tfrac{1}{3}(1-10\varepsilon_s)(|X|+|Y|) + \tfrac{2}{3}(|X|+|Z|+\tfrac{|Y|}{2}) \\ \geq\ & \tfrac{2}{3}(1-10\varepsilon_s)(|X|+|Y|+|Z|) \\ =\ & \tfrac{2}{3}(1-10\varepsilon_s)|M_3|. \end{aligned}$$

This concludes the proof of Lemma 15.

The $\frac{4}{3} + \varepsilon$ approximation mentioned in Theorem 4 is obtained by combining the above approach with some techniques developed for $2DK$ in the weighted case. In particular, we use part of the vertical free strip guaranteed by the resource contraction lemma to pack the *thin* items as defined in that section.

VI. Weighted case with rotations

In the weighted case it is not possible to simply discard large items as this might be too costly. We first show that if there is no *massive* item, i.e., an item with both side lengths at least $(1 - \varepsilon)N$, then we can achieve an analogous resource contraction lemma to get a container packing with a profit of $(\frac{2}{3} - O(\varepsilon))p(OPT)$. We separately consider the case when there exists a massive item and show that even in that case there exists a container packing with $(\frac{2}{3} - O(\varepsilon))p(OPT)$ profit. This gives us a $(\frac{3}{2} + \varepsilon)$-approximation, see Theorem 4.

VII. Open Problems

The main problem that we left open is to find a PTAS, if any, for $2DK$ and $2DKR$. This would be interesting even in the cardinality case. We believe that a better understanding of natural generalizations of L-packings might be useful. For example, is there a PTAS for *ring-packing* instances arising by shifting of long items? This would directly lead to an improved approximation factor for $2DK$ (though not to a PTAS). Is there a PTAS for L-packings *with rotations*? Our improved approximation algorithms for $2DKR$ are indeed based on a different approach. Is there a PTAS for $O(1)$ instances of L-packing? This would also lead to an improved approximation factor for $2DK$, and might be an important step towards a PTAS.

References

[1] J. Y. Leung, T. W. Tam, C. Wong, G. H. Young, and F. Y. Chin, "Packing squares into a square," *Journal of Parallel and Distributed Computing*, vol. 10, no. 3, pp. 271–275, 1990.

[2] K. Jansen and G. Zhang, "On rectangle packing: maximizing benefits," in *SODA*, 2004, pp. 204–213.

[3] A. Adamaszek and A. Wiese, "A quasi-PTAS for the two-dimensional geometric knapsack problem," in *SODA*, 2015, pp. 1491–1505.

[4] N. Bansal, A. Caprara, K. Jansen, L. Prädel, and M. Sviridenko, "A structural lemma in 2-dimensional packing, and its implications on approximability," in *ISAAC*, 2009, pp. 77–86.

[5] A. V. Fishkin, O. Gerber, K. Jansen, and R. Solis-Oba, "Packing weighted rectangles into a square," in *MFCS*, 2005, pp. 352–363.

[6] K. Jansen and R. Solis-Oba, "New approximability results for 2-dimensional packing problems," in *MFCS*, 2007, pp. 103–114.

[7] A. V. Fishkin, O. Gerber, and K. Jansen, "On efficient weighted rectangle packing with large resources," in *ISAAC*, 2005, pp. 1039–1050.

[8] K. Jansen and R. Solis-Oba, "A polynomial time approximation scheme for the square packing problem," in *IPCO*, 2008, pp. 184–198.

[9] S. Heydrich and A. Wiese, "Faster approximation schemes for the two-dimensional knapsack problem," in *SODA*, 2017, pp. 79–98.

[10] A. Adamaszek, T. Kociumaka, M. Pilipczuk, and M. Pilipczuk, "Hardness of approximation for strip packing," *CoRR*, vol. abs/1610.07766, 2016. [Online]. Available: http://arxiv.org/abs/1610.07766

[11] W. Gálvez, F. Grandoni, S. Ingala, and A. Khan, "Improved pseudo-polynomial-time approximation for strip packing," in *FSTTCS*, 2016, pp. 9:1–9:14.

[12] R. Harren, K. Jansen, L. Prädel, and R. van Stee, "A (5/3 + ϵ)-approximation for strip packing," *Comput. Geom.*, vol. 47, no. 2, pp. 248–267, 2014.

[13] K. Jansen and M. Rau, "Improved approximation for two dimensional strip packing with polynomial bounded width," *CoRR*, vol. abs/1610.04430, 2016. [Online]. Available: http://arxiv.org/abs/1610.04430

[14] G. Nadiradze and A. Wiese, "On approximating strip packing with a better ratio than 3/2," in *SODA*, 2016, pp. 1491–1510.

[15] A. Adamaszek and A. Wiese, "Approximation schemes for maximum weight independent set of rectangles," in *FOCS*, 2013, pp. 400–409.

[16] K. Jansen and G. Zhang, "Maximizing the number of packed rectangles," in *SWAT*, 2004, pp. 362–371.

[17] R. Harren, "Approximating the orthogonal knapsack problem for hypercubes," in *ICALP*, 2006, pp. 238–249.

[18] N. Bansal and A. Khan, "Improved approximation algorithm for two-dimensional bin packing," in *SODA*, 2014, pp. 13–25.

[19] A. Khan, "Approximation algorithms for multidimensional bin packing," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, USA, 2016.

[20] N. Bansal, A. Caprara, and M. Sviridenko, "A new approximation method for set covering problems, with applications to multidimensional bin packing," *SIAM Journal on Computing*, vol. 39, no. 4, pp. 1256–1278, 2009.

[21] A. Caprara, "Packing 2-dimensional bins in harmony," in *FOCS*, 2002, pp. 490–499.

[22] C. Kenyon and E. Rémila, "A near-optimal solution to a two-dimensional cutting stock problem," *Mathematics of Operations Research*, vol. 25, no. 4, pp. 645–656, 2000.

[23] F. Chung, M. R. Garey, and D. S. Johnson, "On packing two-dimensional bins," *SIAM Journal on Algebraic Discrete Methods*, vol. 3, pp. 66–76, 1982.

[24] H. I. Christensen, A. Khan, S. Pokutta, and P. Tetali, "Approximation and online algorithms for multidimensional bin packing: A survey," *Computer Science Review*, vol. 24, pp. 63–79, 2017.

[25] A. Steinberg, "A strip-packing algorithm with absolute performance bound 2," *SIAM Journal on Computing*, vol. 26, no. 2, pp. 401–409, 1997.