Check for updates

On the Two-Dimensional Knapsack Problem for Convex Polygons

ARTURO MERINO, Saarland University and Max Planck Institute for Informatics, Saarbrücken, Germany
ANDREAS WIESE, Technical University of Munich, Munich, Germany

We study the two-dimensional geometric knapsack problem for convex polygons. Given a set of weighted convex polygons and a square knapsack, the goal is to select the most profitable subset of the given polygons that fits non-overlappingly into the knapsack. We allow to rotate the polygons by arbitrary angles. We present a quasi-polynomial time $O(1)$-approximation algorithm for the general case and a pseudopolynomial time $O(1)$-approximation algorithm if all input polygons are triangles, both assuming polynomially bounded integral input data. Additionally, we give a quasi-polynomial time algorithm that computes a solution of optimal weight under resource augmentation—that is, we allow to increase the size of the knapsack by a factor of $1+\delta$ for some $\delta > 0$ but compare ourselves with the optimal solution for the original knapsack. To the best of our knowledge, these are the first results for two-dimensional geometric knapsack in which the input objects are more general than axis-parallel rectangles or circles and in which the input polygons can be rotated by arbitrary angles.

CCS Concepts: • **Theory of computation** → **Packing and covering problems;**

Additional Key Words and Phrases: Approximation algorithms, geometric knapsack problem, polygons, rotation

ACM Reference Format:
Arturo Merino and Andreas Wiese. 2024. On the Two-Dimensional Knapsack Problem for Convex Polygons. *ACM Trans. Algor.* 20, 2, Article 16 (April 2024), 38 pages. https://doi.org/10.1145/3644390

1 INTRODUCTION

In the two-dimensional geometric knapsack problem (2DKP), we are given a square knapsack $K := [0, N]\times[0, N]$ for some integer N and a set of n open convex polygons $\mathcal{P}$ where each polygon $P_i \in \mathcal{P}$ has a weight $w_i > 0$; we write $w(\mathcal{P}') := \sum_{P_i \in \mathcal{P}'} w_i$ for any set $\mathcal{P}' \subseteq \mathcal{P}$. We assume that each vertex of each polygon $P \in \mathcal{P}$ has integral coordinates. The goal is to select a subset $\mathcal{P}' \subseteq \mathcal{P}$ of maximum total weight $w(\mathcal{P}')$ such that the polygons in $\mathcal{P}'$ fit non-overlappingly into K if we translate and rotate them suitably (by arbitrary angles). Since the polygons are open sets, we allow their boundaries to overlap in a feasible solution, but we do not allow their interiors to overlap. 2DKP is a natural packing problem. The reader may think of cutting items out of a piece of raw

A. Merino was partially supported by the grants DFG Project 413902284 and ANID Becas Chile 2019-72200522.
This research was carried out while A. Wiese was working at the Universidad de Chile, partially supported by the grants FONDECYT Regular 1170223 and 1200173.
Authors' addresses: A. Merino, Saarland University and Max Planck Institute for Informatics, Saarland Informatics Campus E1.3, 66123 Saarbrücken, Germany; e-mail: merino@cs.uni-saarland.de; A. Wiese, Technical University of Munich, Munich, Boltzmannstraße 3, 85748 Garching bei München, Germany; e-mail: andreas.wiese@tum.de.

ACM 1549-6325/2024/04-ART16
https://doi.org/10.1145/3644390

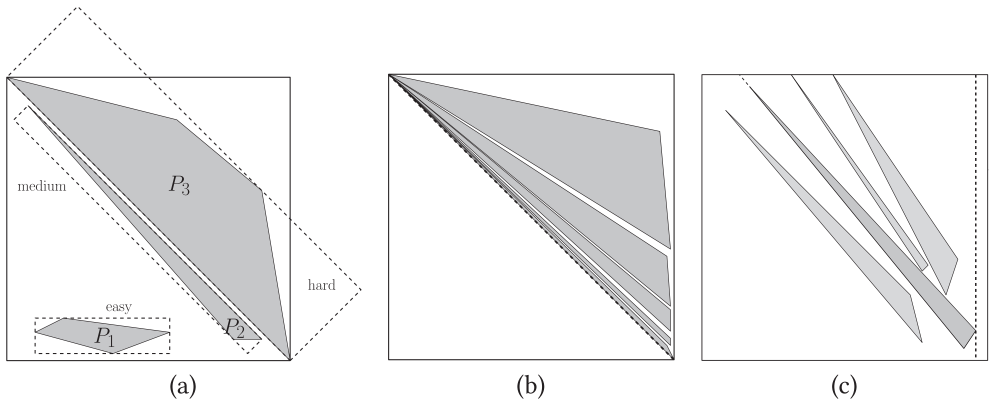


Fig. 1. (a) An easy, a medium, and a hard polygon and their bounding boxes. (b) Triangles packed in a top-left packing. (c) The geometric DP subdivides the knapsack along the dashed lines and then recurses within each resulting area.

material like metal or wood, cutting cookies out of dough, or, in three dimensions, loading cargo into a ship or a truck. In particular, in these applications, the respective items can have various kinds of shapes. Also note that 2DKP is a natural geometric generalization of the classical one-dimensional knapsack problem (see, e.g., other works [11, 15] for the classical one-dimensional version).

Our understanding of 2DKP highly depends on the type and allowed orientation of the input objects. If all polygons are axis-parallel squares, there is a $(1+\epsilon)$-approximation algorithm with a running time of the form $O_\epsilon(1)n^{O(1)}$ (i.e., an EPTAS) [6], and there can be no FPTAS (unless P = NP) since the problem is strongly NP-hard [12]. For axis-parallel rectangles, there is a polynomial time $(17/9+\epsilon) < 1.89$-approximation algorithm and a $(3/2+\epsilon)$-approximation algorithm if the items can be rotated by exactly $\pi/2$ [5]. If the input data is quasi-polynomially bounded, there is a $(1+\epsilon)$-approximation algorithm in quasi-polynomial time [2], with and without the possibility to rotate items by $\pi/2$. For circles, a $(1+\epsilon)$-approximation algorithm is known under resource augmentation in one dimension if the weight of each circle equals its area [13].

To the best of our knowledge, there is no result known for 2DKP for shapes different than axis-parallel rectangles and circles. Additionally, there is no result known in which input polygons are allowed to be rotated by angles different than $\pi/2$. However, in the applications of 2DKP, the items might have shapes that are more complicated than rectangles or circles. Thus, it makes sense to allow rotations by arbitrary angles, such as when cutting items out of some material. In this article, we present the first results for 2DKP in these settings.

1.1 Our Contribution

We study 2DKP for arbitrary convex polygons, allowing to rotate them by arbitrary angles. Note that due to the latter, it might be that some optimal solution places the vertices of the polygons on irrational coordinates, even if all input numbers are integers. Our first results are a quasi-polynomial time $O(1)$-approximation algorithm for general convex polygons and a (respectively) pseudopolynomial time $O(1)$-approximation algorithm for triangles, assuming that N is quasi-polynomially (respectively, polynomially) bounded in the input size.

By rotation, we can assume for each input polygon that the line segment defining its diameter is horizontal. We identify three different types of polygons for which we employ different strategies for packing them (Figure 1(a)). First, we consider the *easy* polygons, which are the polygons whose bounding boxes fit into the knapsack without rotation. We pack these polygons such that their bounding boxes do not intersect. Using area arguments and Steinberg's algorithm [14], we obtain an $O(1)$-approximation for the easy polygons. Then, we consider the *medium* polygons which are the polygons whose bounding boxes easily fit into the knapsack if we can rotate them by $\pi/4$. We use a special type of packing in which the bounding boxes are rotated by $\pi/4$ and then stacked on top of each other (Figure 2(b)). More precisely, we group the polygons by the widths of their bounding boxes, and to each group we assign two rectangular containers in the packing. We compute the essentially optimal solution of this type by solving a generalization of one-dimensional knapsack for each group. Our key structural insight for medium polygons is that such a solution is $O(1)$-approximate. Let OPT be an optimal solution for a given instance of 2DKP. We prove that in OPT the medium polygons of each group occupy an area that is by at most a constant factor bigger than the corresponding containers, and that a constant fraction of these polygons fit into the containers. In particular, we show that medium polygons with very wide bounding boxes lie in a very small hexagonal area close to the diagonal of the knapsack. Our routines for easy and medium polygons run in polynomial time.

It remains to pack the *hard* polygons whose bounding boxes just fit into the knapsack or do not fit at all, even under rotation. Note that this does not imply that the polygon itself does not fit. Our key insight is that there can be only $O(\log N)$ such polygons in the optimal solution. Therefore, we can guess these polygons in quasi-polynomial time, assuming that N is quasi-polynomially bounded. However, in contrast to other packing problems, it is not trivial to check whether a set of given polygons fits into the knapsack since we can rotate them by arbitrary angles and we cannot enumerate all possibilities for the angles. Nevertheless, we show that by losing a constant factor in the approximation guarantee, we can assume that the placement of each hard polygon comes from a pre-computable polynomial size set and hence we can guess the placements of the $O(\log N)$ hard polygons in quasi-polynomial time.

Theorem 1. *There is an $O(1)$-approximation algorithm for 2DKP with a running time of $(nN)^{(\log nN)^{O(1)}}$.*

If all hard polygons are triangles, we present a *pseudopolynomial* time $O(1)$-approximation algorithm. We split the triangles in OPT into two types. For one type, we show that a constant fraction of it can be packed in what we call *top-left-packings* (see Figure 1(b)). In these packings, the triangles are sorted by the lengths of their longest edges and placed on top of each other in a triangular area. We devise a **Dynamic Program (DP)** that essentially computes the most profitable top-left-packing. For proving that this yields an $O(1)$-approximation, we need some careful arguments for rearranging a subset of the triangles with large weight to obtain a packing that our DP can compute. We observe that essentially all hard polygons in OPT must intersect the horizontal line that contains the mid-point of the knapsack. Our key insight is that if we pack a triangle in a top-left-packing, then it intersects this line to a similar extent as in OPT. Then we derive a sufficient condition when a set of triangles fits in a top-left-packing, based on by how much they overlap this line.

For the other type of triangles, we use a geometric DP. In this DP, we recursively subdivide the knapsack into subareas in which we search for the optimal solution recursively (see Figure 1(c)). In the process, we guess the placements of some triangles from OPT. Again, by losing a constant factor, we can assume that for each triangle in OPT there are only a polynomial number of possible placements. By exploiting structural properties of this type of triangles, we ensure that the number of needed DP-cells is bounded by a polynomial. A key difficulty is that we sometimes split the knapsack into two parts on which we recurse independently. Then we need to ensure that we do not select some (possibly high weight) triangle in both parts. To this end, we globally select at most one triangle from each of the $O(\log N)$ groups (losing a constant factor), and when we recurse, we guess for each subproblem from which of the $O(\log N)$ groups it contains a triangle in OPT. This yields only $2^{O(\log N)} = N^{O(1)}$ guesses.

THEOREM 2. *There is an $O(1)$-approximation algorithm for 2DKP with a (pseudopolynomial) running time of $(nN)^{O(1)}$ if all input polygons are triangles.*

Then, we study the setting of resource augmentation—that is, we compute a solution that fits into a larger knapsack of size $(1+\delta)N\times(1+\delta)N$ for some constant $\delta > 0$ and compare ourselves with a solution that fits into the original knapsack of size $N\times N$. We show that then the optimal solution can contain only *constantly* many hard polygons, and hence we can guess them in *polynomial* time.

THEOREM 3. *There is an $O(1)$-approximation algorithm for 2DKP under $(1 + \delta)$-resource augmentation with a running time of $n^{O_\delta(1)}$.*

Finally, we present a quasi-polynomial time algorithm that computes a solution of weight at least $w(\text{OPT})$ (i.e., we do not lose any factor in the approximation guarantee) that is feasible under resource augmentation. This algorithm does not use the preceding classification of polygons into easy, medium, and hard polygons. Instead, we prove that if we can increase the size of the knapsack slightly, we can ensure that for the input polygons there are only $(\log n)^{O_\delta(1)}$ different shapes by enlarging the polygons suitably. Additionally, we show that we need to allow only a polynomial number of possible placements and rotations for each input polygon, *without* sacrificing any polygons from OPT. Then, we use a technique from Adamaszek and Wiese [1] implying that there is a balanced separator for the polygons in OPT with only $(\log n)^{O_\delta(1)}$ edges and which intersects polygons from OPT with only very small area. We guess the separator, guess how many polygons of each type are placed inside and outside the separator, and then recurse on each of these parts. Some polygons are intersected by the balanced separator. However, we ensure that they have very small area in total, and hence we can place them into the additional space of the knapsack that we gain due to the resource augmentation. This generalizes a result in another work by Adamaszek and Wiese [2] for axis-parallel rectangles.

THEOREM 4. *There is an algorithm for 2DKP under $(1 + \delta)$-resource augmentation with a running time of $n^{O_\delta(\log n)^{O(1)}}$ that computes a solution of weight at least $w(\text{OPT})$.*

In our approximation algorithms, we focus on a clean exposition of our methodology for obtaining $O(1)$-approximations, rather than on optimizing the actual approximation ratio.

1.2 Other Related Work

Prior to the results mentioned previously, polynomial time $(2 + \epsilon)$-approximation algorithms for 2DKP for axis-parallel rectangles were presented by Jansen and Zhang [9, 10]. For the same setting, a PTAS is known under resource augmentation in one dimension [7] and there is a polynomial time algorithm computing a solution with optimum weight under resource augmentation in both dimensions [6]. Additionally, there is a PTAS if the weight of each rectangle equals its area [3]. For squares, Jansen and Solis-Oba [8] presented a PTAS.

2 CONSTANT FACTOR APPROXIMATION ALGORITHMS

In this section, we present our quasi-polynomial time $O(1)$-approximation algorithm for general convex polygons and our polynomial time $O(1)$-approximation algorithm for triangles, assuming polynomially bounded input data. Our strategy is to partition the input polygons $\mathcal{P}$ into three classes, easy, medium, and hard polygons, and then to devise algorithms for each class separately.

Let $K := [0, N] \times [0, N]$ denote the given knapsack. We assume that each input polygon is described by the coordinates of its vertices which we assume to be integral. First, we rotate each polygon in $\mathcal{P}$ such that a diametrical segment (i.e., a line segment which joins two of the vertices which are farthest apart) is horizontal. Hence, when we refer to *the* diametrical segment in the following, we mean the diametrical segment which is horizontal. For each polygon $P_i \in \mathcal{P}$, denote by $(x_{i,1}, y_{i,1}), \ldots, (x_{i,k_i}, y_{i,k_i})$ the new coordinates of its vertices. Observe that due to the rotation, the resulting coordinates might not be integral, and possibly not even rational. We will take this into account when we define our algorithms. For each $P_i \in \mathcal{P}$, we define its *bounding box* B_i to be the smallest axis-parallel rectangle that contains P_i. Formally, we define $B_i := [\min_\ell x_{i,\ell}, \max_\ell x_{i,\ell}] \times [\min_\ell y_{i,\ell}, \max_\ell y_{i,\ell}]$. For each polygon P_i, we define ℓ_i as the width of its bounding box and h_i as the height of its bounding box. More formally, $\ell_i := \max_\ell x_{i,\ell} - \min_\ell x_{i,\ell}$ and $h_i := \max_\ell y_{i,\ell} - \min_\ell y_{i,\ell}$. Note that ℓ_i is also the length of the diametrical segment of P_i (i.e., the diameter of P_i). If necessary, we will work with suitable estimates of these values later, considering that they might be irrational and hence we may not compute them exactly.

We first distinguish the input polygons into *easy*, *medium*, and *hard* polygons. We say that a polygon P_i is *easy* if B_i fits into K without rotation (i.e., such that $\ell_i \leq N$ and $h_i \leq N$). Denote by $\mathcal{P}_E \subseteq \mathcal{P}$ the set of easy polygons. Note that the bounding box of a polygon in $\mathcal{P} \setminus \mathcal{P}_E$ might still fit into K if we rotate it suitably. Intuitively, we will define the medium polygons to be the polygons P_i whose bounding box B_i fits into K with quite some slack if we rotate B_i properly and the hard polygons are the remaining polygons (in particular, those polygons whose bounding box does not fit at all into K).

Formally, for each polygon $P_i \in \mathcal{P} \setminus \mathcal{P}_E$, we define $h'_i := \sqrt{2}N - \ell_i$. The intuition for h'_i is that a rectangle of width ℓ_i and height h'_i is the highest rectangle of width ℓ_i that still fits into K after rotation.

Lemma 5. *Let $P_i \in \mathcal{P} \setminus \mathcal{P}_E$. A rectangle of width ℓ_i and height h'_i fits into K (if we rotate it by $\pi/4$), but a rectangle of width ℓ_i and of height larger than h'_i does not fit into K.*

Proof. First observe that since P_i is not easy, we have that $\ell_i > N$. A simple computation shows that $R_i = [0, \ell_i] \times [0, h'_i]$ fits into K when rotating by $\pi/4$ with new vertices at $(\frac{h'_i}{\sqrt{2}}, 0)$, $(N, \frac{\ell_i}{\sqrt{2}})$, $(N - \frac{h'_i}{\sqrt{2}}, N)$, $(0, N - \frac{\ell_i}{\sqrt{2}})$.

We now prove the second part of the lemma. Suppose $R^*_i = [0, \ell_i] \times [0, h^*]$ is placed into the knapsack after some rotation. We call the bottom edge of R^*_i (which has length ℓ_i) the *base* of R^*_i. W.l.o.g., we can assume that the base of R^*_i intersects the boundary of K at exactly two points p_1 and p_2 (otherwise, we shift the base of R^*_i toward the boundary of the knapsack). Since $\ell_i > N$, we can split K along the base of R^*_i into two polygons and one of the polygons will be a triangle. Let $\alpha > 0$ be the smallest angle of this triangle. Define two parallel lines L_1 and L_2 that intersect p_1 and p_2, respectively, and are perpendicular to the base of R^*_i. Note that R^*_i is contained entirely between L_1 and L_2. Thus, the height of R^*_i is bounded by the minimum of the lengths of the line segments of L_1 and L_2 that are contained within K. This minimum is maximized when both line segments have equal length. Indeed, suppose for the sake of contradiction that L_1 is smaller than L_2. Then, we rotate the base, making α larger such that L_1 is still at most as large as L_2, while the length of L_1 increases. Since both line segments have equal length when $\alpha = \pi/4$, we obtain that $h^* \leq h'_i$. □

Hence, if h_i is much smaller than h'_i, then B_i fits into K with quite some slack. Therefore, we define that a polygon $P_i \in \mathcal{P} \setminus \mathcal{P}_E$ is *medium* if $h_i \leq h'_i/8$ and *hard* otherwise. Denote by $\mathcal{P}_M \subseteq \mathcal{P}$ and $\mathcal{P}_H \subseteq \mathcal{P}$ the medium and hard polygons, respectively. We will present $O(1)$-approximation algorithms for each of the sets $\mathcal{P}_E, \mathcal{P}_M, \mathcal{P}_H$ separately. The best of the computed sets will then yield an $O(1)$-approximation overall.

For the easy polygons, we construct a polynomial time $O(1)$-approximation algorithm in which we select polygons such that we can pack their bounding boxes as non-overlapping rectangles using Steinberg's algorithm [4] (see Section 2.1). The approximation ratio follows from area arguments.

Lemma 6. *There is an algorithm with a running time of* $n^{O(1)}$ *that computes a solution* $\mathcal{P}'_E \subseteq \mathcal{P}_E$ *with* $w(\mathrm{OPT} \cap \mathcal{P}_E) = O(w(\mathcal{P}'_E))$.

For the medium polygons, we obtain an $O(1)$-approximation algorithm using a different packing strategy (see Section 2.2).

Lemma 7. *There is an algorithm with a running time of* $n^{O(1)}$ *that computes a solution* $\mathcal{P}'_M \subseteq \mathcal{P}_M$ *with* $w(\mathrm{OPT} \cap \mathcal{P}_M) = O(w(\mathcal{P}'_M))$.

The most difficult polygons are the hard polygons. First, we show that in quasi-polynomial time, we can obtain an $O(1)$-approximation for them (see Section 2.3).

Lemma 8. *There is an algorithm with a running time of* $(nN)^{(\log nN)^{O(1)}}$ *that computes a solution* $\mathcal{P}'_H \subseteq \mathcal{P}_H$ *with* $w(\mathrm{OPT} \cap \mathcal{P}_H) = O(w(\mathcal{P}'_H))$.

Combining Lemmas 6 through 8 yields Theorem 1. Furthermore, if all polygons are triangles, we obtain an $O(1)$-approximation in polynomial time. The following lemma is proved in Section 2.4 and together with Lemmas 6 and 7 implies Theorem 2.

Lemma 9. *If all input polygons are triangles, then there is an algorithm with a running time of* $(nN)^{O(1)}$ *that computes a solution* $\mathcal{P}'_H \subseteq \mathcal{P}_H$ *with* $w(\mathrm{OPT} \cap \mathcal{P}_H) = O(w(\mathcal{P}'_H))$.

Orthogonal to the characterization into easy, medium, and hard polygons, we subdivide the non-easy polygons further into classes according to their diameter. This will prove useful in the development of our constant approximation algorithms. More precisely, we do this according to their difference between ℓ_i and the diameter of K (i.e., $\sqrt{2}N$). Formally, we introduce the following definition.

Definition 2.1 (Classification of Polygons by Diagonals). For each $j \in \mathbb{Z}$, we define

$$\mathcal{P}_j := \left\{ P_i \in \mathcal{P} \setminus \mathcal{P}_E \mid \ell_i \in \left[\sqrt{2}N - 2^j, \sqrt{2}N - 2^{j-1} \right) \right\}.$$

We note that every non-easy polygon belongs to some set $\mathcal{P}_j$, except for the polygons with diameter *exactly* $\sqrt{2}N$. For every packing of such a polygon P, the midpoint of the knapsack must be contained in the boundary of P. Therefore, any feasible solution to 2DKP contains at most two such polygons. Hence, if such polygons are contained in OPT and constitute a constant fraction of the profit of OPT, we obtain an $O(1)$-approximate solution easily by simply guessing the most profitable such polygon. Thus, in our approximation results, we assume that there are no polygons with diameter exactly $\sqrt{2}N$ and this loses only a constant factor in the approximation ratio. Additionally, we note that only finitely many $\mathcal{P}_j$ are non-empty. To this end, we define $j_{\min} := -\lceil \log N \rceil - 1$ and $j_{\max} := 1 + \lfloor \log((\sqrt{2} - 1)N) \rfloor$ and show that these are the minimum and maximum values of j such that $\mathcal{P}_j$ is non-empty.

Lemma 10. *If* $\mathcal{P}_j \neq \emptyset$, *then* $j \in \{j_{\min}, \dots, j_{\max}\}$.

Proof. Assume that j is an integer with $j > j_{\max} = 1 + \lfloor \log((\sqrt{2} - 1)N) \rfloor$. We want to show that then $\mathcal{P}_j = \emptyset$. In particular, then $j \geq 2 + \lfloor \log((\sqrt{2} - 1)N) \rfloor$ and thus $\sqrt{2}N - 2^{j-1} \leq \sqrt{2}N -$

$2^{1+\lfloor \log((\sqrt{2}-1)N) \rfloor} \leq \sqrt{2}N - 2^{\log((\sqrt{2}-1)N)} = N$. Thus, for every polygon $P_i \in \mathcal{P}_j$, we have $\ell_i \leq \sqrt{2}N - 2^{j-1} \leq N$. We conclude that $P_i \in \mathcal{P}_E$ and, therefore, $P_i \notin \mathcal{P}_j$ and $\mathcal{P}_j = \emptyset$.

Assume now that j is an integer with $j < j_{\min} = -\lceil \log N \rceil - 1$ and hence $j \leq -\lceil \log N \rceil - 2$. We want to argue that then $\mathcal{P}_j = \emptyset$. Note that ℓ_i^2 must be a positive integer. Thus, it suffices to argue that $[(\sqrt{2}N - 2^j)^2, (\sqrt{2}N - 2^{j-1})^2) \cap \mathbb{Z} = \emptyset$. We prove that $[(\sqrt{2}N - 2^j)^2, (\sqrt{2}N)^2) \cap \mathbb{Z} = \emptyset$, which directly implies the result. This is the case since

$$\left(\sqrt{2}N\right)^2 - \left(\sqrt{2}N - 2^j\right)^2 = 2^j \left(2\sqrt{2}N - 2^j\right) \leq 2^j 2\sqrt{2}N \leq 2^{-\lceil \log N \rceil - 2} 2\sqrt{2}N < 1. \qquad \square$$

Furthermore, for each polygon $P_i \in \mathcal{P} \setminus \mathcal{P}_E$, we can compute its group $\mathcal{P}_j$ even though ℓ_i might be irrational.

2.1 Easy Polygons

We present an $O(1)$-approximation algorithm for the polygons in $\mathcal{P}_E$. First, we show that the area of each polygon is at least half of the area of its bounding box. We will use this later for defining lower bounds using area arguments. We also introduce the following notation: for a set $S \subseteq K$, we use conv(S) to denote the convex hull of S, and for a set $O \subseteq \mathbb{R}^2$, we denote its area by area(O).

Lemma 11. *For each $P_i \in \mathcal{P}$, it holds that* $\text{area}(P_i) \geq \frac{1}{2}\text{area}(B_i)$.

Proof. Let D be the diametrical segment, and recall that we rotated P_i such that D was horizontal. We split P_i into a polygon P which lies above D and a polygon Q which lies below D. Let p_P be a point with maximum y-coordinate in P_i and p_Q be a point with minimum y-coordinate in P_i. We define T_P and T_Q as $\text{conv}(D \cup \{p_P\})$ and $\text{conv}(D \cup \{p_Q\})$, respectively. Note that T_P and T_Q are either triangles or line segments. Furthermore, by convexity, we know that $T_P \subseteq P$ and $T_Q \subseteq Q$. The lemma then follows by noting that $\frac{1}{2}\text{area}(B_i) = \text{area}\left(T_P \cup T_Q\right) \leq \text{area}(P \cup Q) = \text{area}(P_i)$. $\square$

However, it is known that we can pack any set of axis-parallel rectangles into K, as long as their total area is at most area$(K)/2$ and each single rectangle fits into K without rotation. This is implied by the following theorem.

Theorem 12 ([14]). *Let $r_1, \ldots, r_k$ be a set of axis-parallel rectangles such that $\sum_{i=1}^{k} \text{area}(r_i) \leq \text{area}(K)/2$ and each individual rectangle r_i fits into K. Then, there is a polynomial time algorithm that packs $r_1, \ldots, r_k$ into K.*

We first compute (essentially) the most profitable set of polygons from $\mathcal{P}_E$ whose total area is at most area(K) via a reduction to one-dimensional knapsack.

Lemma 13. *In time $(\frac{n}{\epsilon})^{O(1)}$, we can compute a set of polygons $\mathcal{P}' \subseteq \mathcal{P}_E$ such that $w(\mathcal{P}') \geq (1 - \epsilon)w(\text{OPT} \cap \mathcal{P}_E)$ and $\sum_{P_i \in \mathcal{P}'} \text{area}(P_i) \leq \text{area}(K)$.*

Proof. We define an instance of one-dimensional knapsack with a set of items I where we introduce for each polygon $P_i \in \mathcal{P}_E$ an item $a_i \in I$ with size $s_i := \text{area}(P_i)$ and profit w_i and define the size of the knapsack to be area(K). We invoke an arbitrary FPTAS for one-dimensional knapsack to solve this instance (e.g., the recent algorithm of Jin [11]) and obtain a set of items $I' \subseteq I$ such that $w(I') \geq (1 - \epsilon)\text{OPT}(I)$, where OPT$(I)$ denotes the optimal solution for the set of items I, given a knapsack of size area(K). We define $\mathcal{P}' := \{P_i \in \mathcal{P}_E \mid a_i \in I'\}$ and note that it fulfills the statement of the lemma. $\square$

The idea is now to partition $\mathcal{P}'$ into at most seven sets $\mathcal{P}'_1, \ldots, \mathcal{P}'_7$. Hence, one of these sets must contain at least a profit of $w(\mathcal{P}')/7$. We will define this partition such that each set $\mathcal{P}'_j$ contains only one polygon or its polygons have a total area of at most area$(K)/4$.

Lemma 14. *Given a set $\mathcal{P}' \subseteq \mathcal{P}_E$ with $\sum_{P_i \in \mathcal{P}'} \text{area}(P_i) \leq \text{area}(K)$. In polynomial time, we can compute a set $\mathcal{P}'' \subseteq \mathcal{P}'$ with $w(\mathcal{P}'') \geq \frac{1}{7} w(\mathcal{P}')$ and additionally $\sum_{P_i \in \mathcal{P}''} \text{area}(P_i) \leq \text{area}(K)/4$ or $|\mathcal{P}''| = 1$.*

Proof. If $|\mathcal{P}'| \leq 3$, then the claim is trivial. Otherwise, we partition the items in $\mathcal{P}'$ greedily into four sets C_1, C_2, C_3, C_4 such that for each set C_j with $j \in \{1, 2, 3\}$ there is a polygon $P_j \in C_j$ such that the polygons in $C_j \setminus \{P_j\}$ have a total area of at most area$(K)/4$. Furthermore, C_4 contains items with a total area of at most area$(K)/4$. To obtain these sets, we run a Next-Fit type algorithm which inserts the polygons into bins and closes a bin once the area of the polygons in this bin is strictly larger than area$(K)/4$. We start by inserting the polygons in $\mathcal{P}'$ into C_1 until the total area of the polygons in C_1 is strictly larger than area$(K)/4$. Then, we insert the following polygons from $\mathcal{P}'$ into C_2 until the total area of the polygons in C_2 is strictly larger than area$(K)/4$ and so forth. Finally, one of the seven sets $C_1 \setminus \{P_1\}, \{P_1\}, C_2 \setminus \{P_2\}, \{P_2\}, C_3 \setminus \{P_3\}, \{P_3\}, C_4$ fulfills the claim. □

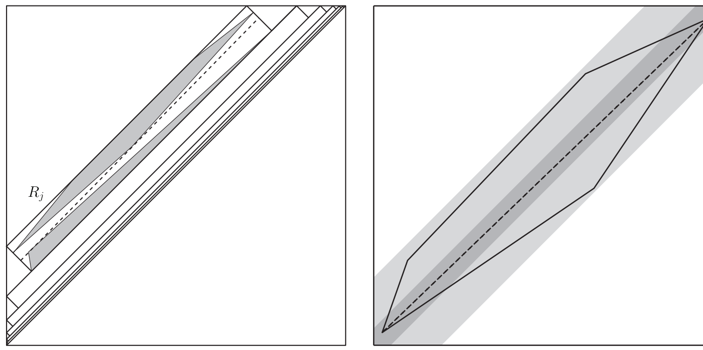
Note that any set of rectangles with total area at most $\frac{1}{2}$area(K) can be packed into the knapsack by Theorem 12. Therefore, we can pack any set of easy polygons with total area at most $\frac{1}{4}$area(K) by first putting each polygon into its bounding box, then packing these bounding boxes (having a total area of at most $\frac{1}{2}$area(K)) using Theorem 12. Moreover, if the height and width of the bounding box can be computed in polynomial time, this placement can also be computed in polynomial time. If $|\mathcal{P}''| = 1$, we simply pack the single polygon in $\mathcal{P}''$ into the knapsack. Otherwise, using Lemmas 11 and 13 and Theorem 12, we know that we can pack the bounding boxes of the polygons in $\mathcal{P}''$ into K. Note that their heights and widths might be irrational. Therefore, we slightly increase them such that these values become rational, before applying Theorem 12 to compute the actual packing. If as a result the total area of the bounding boxes exceeds area$(K)/2$, we partition them into two sets where each set satisfies that the total area of the bounding boxes is at most area$(K)/2$ or it contains only one polygon, and we keep the more profitable of these two sets (hence losing a factor of 2 in the approximation ratio). This yields an $O(1)$-approximation algorithm for the easy polygons and thus proves Lemma 6.

2.2 Medium Polygons

We describe an $O(1)$-approximation algorithm for the polygons in $\mathcal{P}_M$. Recall that $\mathcal{P}_j = \{i \in \mathcal{P} \setminus \mathcal{P}_E | \ell_i \in [\sqrt{2}N - 2^j, \sqrt{2}N - 2^{j-1})\}$. Furthermore, by Lemma 10, $\mathcal{P}_j \neq \emptyset$ implies that $j \in \{j_{\min}, \ldots, j_{\max}\}$. For each integer $j \in \{j_{\min}, \ldots, j_{\max}\}$, we define a rectangular container R_j for polygons in $\mathcal{P}_M \cap \mathcal{P}_j$, each container having width $\sqrt{2}N - 2^{j-1}$ and height 2^{j-3} (see Figure 2(a)). Let $\mathcal{R}$ be the set of all containers (i.e., $\mathcal{R} := \{R_j : j \in \{j_{\min}, \ldots, j_{\max}\}\}$). First, we show that we can pack all containers in $\mathcal{R}$ into K (if we rotate them by $\pi/4$).

Lemma 15. *The rectangles in $\mathcal{R}$ can be packed non-overlappingly into K.*

Proof. We describe how we place the rectangles in $\mathcal{R}$ inside K. For each $j \in \{j_{\min}, \ldots, j_{\max}\}$, we place a vertex at the top edge of K at the coordinate $(N - \frac{2^{j-1}}{\sqrt{2}}, N)$ and a vertex at the left edge of K at the coordinate $(0, \frac{2^{j-1}}{\sqrt{2}})$. Then we connect the two vertices corresponding to each j by a line segment. This yields containers with the shapes of trapezoids inside which we place the rectangles in $\mathcal{R}$. More formally, for each $j \in \{j_{\min}, \ldots, j_{\max}\}$, we place a rectangle R_j such that its vertices are at the coordinates $(N - \frac{2^{j-1}}{\sqrt{2}}, N)$, $(0, \frac{2^{j-1}}{\sqrt{2}})$, $(\frac{2^{j-3}}{\sqrt{2}}, \frac{3}{\sqrt{2}} 2^{j-3})$, $(N - \frac{3}{\sqrt{2}} 2^{j-3}, N - \frac{2^{j-3}}{\sqrt{2}})$. Hence, it has a width of $\|(N - \frac{2^{j-1}}{\sqrt{2}}, N) - (0, \frac{2^{j-1}}{\sqrt{2}})\| = \sqrt{2}N - 2^{j-1}$ and a height of $\|(\frac{2^{j-3}}{\sqrt{2}}, \frac{3}{\sqrt{2}} 2^{j-3}) - (0, \frac{2^{j-1}}{\sqrt{2}})\| = 2^{j-3}$,



(a) (b)

Fig. 2. (a) The containers for the medium polygons of the different groups. Within each container, the polygons are stacked on top of each other such that their respective bounding boxes do not intersect. (b) Assume that the polygon (black line segments) is a medium polygon contained in the set $\mathcal{P}_j$. Then, the diametrical segment (dashed) must lie into the dark gray area and the whole polygon must be contained in the light gray area.

as required. A simple angle computation then shows that these are the coordinates of the vertices of R_i when rotated by $\pi/4$. □

For each $j \in \{j_{\min}, \ldots, j_{\max}\}$, we will compute a set of polygons $\mathcal{P}'_j \subseteq \mathcal{P}_M \cap \mathcal{P}_j$ of large weight. Within each container R_j, we will stack the bounding boxes of the polygons in $\mathcal{P}'_j$ on top of each other and then place the polygons in $\mathcal{P}'_j$ in their respective bounding boxes (see Figure 2(a)). In particular, a set of items $\mathcal{P}''_j \subseteq \mathcal{P}_j$ fits into R_j using this strategy if and only if $h(\mathcal{P}''_j) := \sum_{P_i \in \mathcal{P}''_j} h_i \leq 2^{j-3}$.[1] We compute the essentially most profitable set of items $\mathcal{P}'_j$ that fits into R_j with the preceding strategy. For this, we need to solve a one-dimensional knapsack problem that represents filling the height of R_j. The value h_i for a polygon P_i might be irrational, so we work with a $(1+\epsilon)$-estimate of h_i instead. This costs only a factor $O(1)$ in the approximation guarantee. Furthermore, since these polygons are *medium* polygons, they still fit into the knapsack with some slack.

Lemma 16. *Let $\epsilon > 0$. For each $j \in \mathbb{Z}$, there is an algorithm with a running time of $\left(\frac{n}{\epsilon}\right)^{O(1)}$ that computes a set $\mathcal{P}'_j \subseteq \mathcal{P}_j \cap \mathcal{P}_M$ such that $h(\mathcal{P}'_j) \leq 2^{j-3}$ and $w(\mathcal{P}^*{}_j) = O(w(\mathcal{P}'_j))$ for any set $\mathcal{P}^*_j \subseteq \mathcal{P}_j \cap \mathcal{P}_M$ such that $h(\mathcal{P}^*_j) \leq 2^{j-3}$.*

Proof. For each j, we need to solve an instance of the one-dimensional knapsack problem. Here, we have a knapsack, each with capacity 2^{j-3}, the set of objects to choose from are $\mathcal{I} := \mathcal{P}_j \cap \mathcal{P}_M$, and each $i \in \mathcal{I}$ has size h_i and profit w_i. We invoke an arbitrary FPTAS for one-dimensional knapsack to solve this instance (e.g., the recent algorithm of Jin [11]). □

For each $j \in \mathbb{Z}$ with $\mathcal{P}_j \cap \mathcal{P}_M \neq \emptyset$, we apply Lemma 16 and obtain a set $\mathcal{P}'_j$. We pack $\mathcal{P}'_j$ into R_j using that $h(\mathcal{P}'_j) \leq h(R_j)$. Then, we pack all containers R_j for each $j \in \mathbb{Z}$ into K, using Lemma 15.

Let $\mathcal{P}'_M := \bigcup_j \mathcal{P}'_j$ denote the selected polygons. We want to show that $\mathcal{P}'_M$ has large weight. More precisely, we want to show that $w(\text{OPT} \cap \mathcal{P}_M) = O(w(\mathcal{P}'_M))$. First, for each $j \in \mathbb{Z}$, we bound the area of the polygons in $\mathcal{P}_j \cap \mathcal{P}_M \cap \text{OPT}$. To this end, we show that they are contained inside

[1]Observe that for a polygon $P_i \in \mathcal{P}_j$ with $P_i \in \mathcal{P}_H$, it is not necessarily true that $h_i \leq 2^{j-3}$. Hence, this strategy is not suitable for hard polygons.

a certain (irregular) hexagon (see Figure 2) which has small area if the polygons $P_i \in \mathcal{P}_j$ are wide (i.e., if ℓ_i is close to $\sqrt{2}N$). The reason is that then P_i must be placed close to the diagonal of the knapsack and otherwise h_i is relatively small (since P_i is medium), which implies that all of P_i lies close to the diagonal of the knapsack.

Lemma 17. *For each j, it holds that* $\text{area}(\mathcal{P}_j \cap \mathcal{P}_M \cap \text{OPT}) = O(\text{area}(R_j))$.

We defer the proof of Lemma 17 to Section 2.2.1. Using this, we can partition $\mathcal{P}_j \cap \mathcal{P}_M \cap \text{OPT}$ into $O(1)$ subsets such that for each subset $\mathcal{P}'$ it holds that $h(\mathcal{P}') \leq 2^{j-3}$ and hence $\mathcal{P}'$ fits into R_j using our preceding packing strategy. Here, we use that each medium polygon $P_i \in \mathcal{P}_j$ satisfies that $h_i \leq 2^{j-3}$.

Lemma 18. *For each $j \in \{j_{\min} \ldots, j_{\max}\}$, there is a set $\mathcal{P}_j^* \subseteq \mathcal{P}_j \cap \mathcal{P}_M \cap \text{OPT}$ with $w(\mathcal{P}_j \cap \mathcal{P}_M \cap \text{OPT}) = O(w(\mathcal{P}_j^*))$ such that $h(\mathcal{P}_j^*) \leq 2^{j-3}$.*

Proof. We follow a Next-Fit strategy, similarly to the Next-Fit algorithm for bin packing. We begin by partitioning $\mathcal{P}_j \cap \mathcal{P}_M \cap \text{OPT} = \{P'_1, \ldots, P'_m\}$ into groups such that each group fits into R_j. To do so, define a sequence $s_0, s_1, \ldots, s_p$ recursively such that

$$s_0 = 0,$$
$$s_\ell = \max\left\{k \in \{s_{\ell-1}+1, \ldots, m\} \,\middle|\, \sum_{i=s_{\ell-1}+1}^{k} h(P'_i) \leq 2^{j-3}\right\}, \text{ if } s_{\ell-1} < m.$$

Consider the partition of $\mathcal{P}_j \cap \mathcal{P}_M \cap \text{OPT}$ into p parts of the form $C_\ell = \{P'_{s_\ell+1}, \ldots, P'_{s_{\ell+1}}\}$. For $1 \leq \ell < p$, we have $\sum_{P_i \in C_\ell} h_i \geq \left(1 - \frac{1}{8}\right)2^{j-3}$, since $h_i \leq \frac{h'_i}{8} \leq \frac{1}{8}2^{j-3}$. Therefore,

$$\sum_{P_i \in C_\ell} \text{area}(P_i) \geq \frac{1}{2}\sum_{P_i \in C_\ell} \text{area}(B_i) = \frac{1}{2}\sum_{P_i \in C_\ell} h_i \ell_i \geq \frac{1}{2}\left(1 - \frac{1}{8}\right)2^{j-3}\left(\sqrt{2}N - 2^j\right).$$

Moreover, as the polygons are not in $\mathcal{P}_E$, we get that $\sqrt{2}N - 2^{j-1} > N$. Thus,

$$\sqrt{2}N - 2^j > \left(2 - \sqrt{2}\right)N \geq \left(2 - \sqrt{2}\right)N\left(1 - \frac{2^{j-1}}{\sqrt{2}N}\right) = \left(\sqrt{2} - 1\right)\left(\sqrt{2}N - 2^{j-1}\right) > \frac{1}{7}\left(\sqrt{2}N - 2^{j-1}\right).$$

Using this, we obtain

$$\sum_{P_i \in C_\ell} \text{area}(P_i) \geq \frac{7}{16}2^{j-3}\left(\sqrt{2}N - 2^j\right) \geq \frac{1}{16}2^{j-3}\left(\sqrt{2}N - 2^{j-1}\right) = \frac{1}{16}\text{area}(R_j).$$

By Lemma 17, we deduce that there is a constant $M > 0$ such that $\text{area}(\mathcal{P}_j \cap \mathcal{P}_M \cap OPT) \leq M \cdot \text{area}(R_j)$. Hence,

$$M \cdot \text{area}(R_j) \geq \text{area}(\mathcal{P}_j \cap \mathcal{P}_M \cap OPT) \geq \sum_{\ell=1}^{p-1}\sum_{P_i \in C_\ell} \text{area}(P_i) \geq \frac{p-1}{16}\text{area}(R_j),$$

concluding that $p \leq 16M + 1 \leq 17M$. Call C^* the most profitable set of our partition. Using our bound on p, we conclude that

$$w(\mathcal{P}_j \cap \mathcal{P}_M \cap OPT) \leq p \cdot w(C^*) \leq 17M \cdot w(C^*). \qquad \square$$

By combining Lemmas 15, 16, and 18, we obtain the proof of Lemma 7.

2.2.1 Area Bound for Medium Polygons. In this section, we prove Lemma 17. We show that for every $j \in \{j_{\min}, \ldots, j_{\max}\}$, we can choose C_j, independent of i, such that if a polygon $P_i \in \mathcal{P}_j \cap \mathcal{P}_M$ is placed in the knapsack, then it is contained in the (irregular) hexagon $H_{1,j}$ with vertices $(0,0)$, $(0, C_j)$, $(N, N - C_j)$, (N, N), $(N - C_j, N)$, $(C_j, 0)$, or in the (irregular) hexagon $H_{2,j}$ with vertices $(0, N - C_j)$, $(0, N)$, (C_j, N), (N, C_j), $(N, 0)$, $(N - C_j, 0)$; Figure 2(b) presents an illustration of $H_{2,j}$. To prove this, we first show that in the placement of P_i inside K, the vertices of the diametrical segment of P_i essentially lie near opposite corners of K. We then use this to show that this diametrical segment of P_i lies inside one of two hexagons that are even smaller than $H_{1,j}$ and $H_{2,j}$, respectively (see Figure 2(b)). If additionally $P_i \in \mathcal{P}_M$, we conclude that P_i is placed completely within $H_{1,j} \cup H_{2,j}$ (whereas if $P_i \in \mathcal{P}_H$, the latter is not necessarily true). Given a value $r \geq 0$, we define $T_1(r)$ as the union of the two triangles $(0, r), (r, 0), (0, 0)$ and $(N - r, N), (N, N - r), (N, N)$. Similarly, we define $T_2(r)$ as the union of the two triangles $(0, N - r), (0, N), (r, N)$ and $(N - r, 0), (N, 0), (N, r)$. We now define the following two hexagons $H_1(r) := \text{conv}(T_1(r))$ and $H_2(r) := \text{conv}(T_2(r))$. Note that $H_{1,j} = H_1(C_j)$ and $H_{2,j} = H_2(C_j)$.

For any $P_i \in \mathcal{P}$, we define $r_i := N - \sqrt{\ell_i^2 - N^2}$.

Claim 1. *Consider a polygon $P_i \in \mathcal{P}_j$. Let P be a placement of P_i inside K, and let $D = \overline{v^1 v^2}$ denote the diametrical segment of P.*

It holds that (1) $\{v^1, v^2\} \subseteq T_1(r_i)$ or $\{v^1, v^2\} \subseteq T_2(r_i)$, and (2) $D \subseteq H_1(r_i)$ or $D \subseteq H_2(r_i)$. If additionally $P_i \in \mathcal{P}_M$, then $P \subseteq H_1(r_i + \sqrt{2}2^{j-3})$ or $P \subseteq H_2(r_i + \sqrt{2} \cdot 2^{j-3})$.

Proof of claim. We define S as $K \setminus (T_1(r_i) \cup T_2(r_i))$. We first show that $\text{diam}(S) < \ell_i$. To do this, we distinguish two cases: when $N < \ell_i \leq \frac{\sqrt{5}}{2}N$ and when $\ell_i > \frac{\sqrt{5}}{2}N$. If $N \leq \ell_i \leq \frac{\sqrt{5}}{2}N$, then

$$r_i = N - \sqrt{\ell_i^2 - N^2} \geq N - N/2 = N/2,$$

which implies that $r_i \geq N - r_i$ and consequently S is a square of diagonal $2(N - r_i) \leq N < \ell_i$. We conclude that in this case $\text{diam}(S) < \ell_i$.

Suppose now that $\ell_i > \frac{\sqrt{5}}{2}N$ (see Figure 3 for an illustration). This implies that

$$r_i = N - \sqrt{\ell_i^2 - N^2} < N - N/2 = N/2,$$

and consequently $r_i < N - r_i$ (Figure 3). In this case, S is an octagon with vertices $(0, r_i), (r_i, 0), (N - r_i, N), (N, N - r_i), (0, N - r_i), (r_i, N), (N - r_i, 0), (N, r_i)$. Hence,

$$\text{diam}(S) = \sqrt{N^2 + (N - r_i)^2} \leq \sqrt{N^2 + (\ell_i^2 - N^2)} = \ell_i.$$

Since $\text{diam}(S) \leq \ell_i$, we have that $v^1 \in T_1(r_i) \cup T_2(r_i)$ or $v^2 \in T_1(r_i) \cup T_2(r_i)$. Assume w.l.o.g. that $v^1 \in T_1(r_i) \cup T_2(r_i)$. Furthermore, assume w.l.o.g. that $v^1 \in T_1(r_i)$ and that v^1 is contained in the triangle with vertices $(0, r_i), (r_i, 0), (0, 0)$. Furthermore, we have that v^2 is not inside $K \setminus T^1(r_i)$. Otherwise, we have that

$$\ell_i < \sqrt{N^2 + (N - r_i)^2} = \ell_i.$$

Therefore, $v^2 \in T_1(r_i)$ and consequently $D \subseteq H_1(r_i)$. Assume now additionally that $P_i \in \mathcal{P}_M$. Let v be a vertex of P with $v^1 \neq v \neq v^2$. The distance between v and D is at most $h_i \leq \frac{1}{8}(\sqrt{2}N - \ell_i) \leq 2^{j-3}$. Therefore, v is contained in $H_1(r_i + \sqrt{2} \cdot 2^{j-3})$ and hence $P \subseteq H_1(r_i + \sqrt{2} \cdot 2^{j-3})$. The case $v^1 \in T_2(r_i)$ can be handled similarly, and in this case we conclude that $v^2 \in T_2(r_i)$, $D \subseteq H_2(r_i)$, and $P \subseteq H_2(r_i + \sqrt{2} \cdot 2^{j-3})$. □

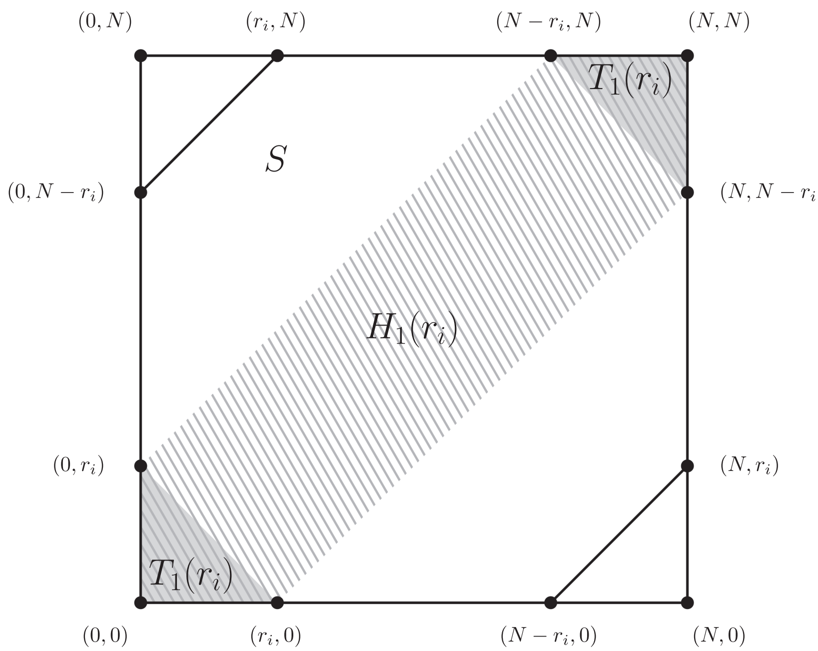


Fig. 3. Some of the elements that appear in the proof for the area bound of medium polygons. The white area denotes S, which is an octagon if $r_i < N - r_i$. The dashed dark gray area is the hexagon $H_1(r_i)$. The two light gray triangles are $T_1(r_i)$.

Since $P_i \in \mathcal{P}_j \cap \mathcal{P}_M$ implies that $j \in \{j_{\min}, \ldots, j_{\max}\}$ by Lemma 10. We deal with $j = j_{\max}$ separately. Here, we define $C_{j_{\max}} := N$ and note that $H_1(C_j)$ is the complete knapsack. Furthermore,

$$\text{area}(R_j) = \left(\sqrt{2}N - 2^{\lfloor \log(\sqrt{2}-1)N \rfloor}\right) 2^{\lfloor \log(\sqrt{2}-1)N \rfloor - 2} \geq \frac{\sqrt{2}-1}{8} N^2 = \frac{\sqrt{2}-1}{8} \text{area}(H_1(C_{j_{\max}})),$$

so the lemma holds for $j_{\max}$.

Hence, we only deal with the case $j \in \{j_{\min}, \ldots, j_{\max}-1\}$. In this case, it holds that $\sqrt{2}N - 2^j \geq N$. Thus, we define

$$C_j := N - \sqrt{\left(\sqrt{2}N - 2^j\right)^2 - N^2} + \sqrt{2} \cdot 2^{j-3}.$$

Note that for every placement P of P_i, we have that $P \subseteq H_1(r_i + \sqrt{2} \cdot 2^{j-3}) \subseteq H_1(C_j) = H_{1,j}$ or $P \subseteq H_2(r_i + \sqrt{2} \cdot 2^{j-3}) \subseteq H_2(C_j) = H_{2,j}$ as $r_i \leq N - \sqrt{(\sqrt{2}N - 2^j)^2 - N^2}$.

Note that we can compute area$(H_1(r))$ by computing the area of half the hexagon and then multiplying by 2—that is, by computing the area of the quadrilateral $(0,0)$, (N,N), $(N-r,N)$, $(0,r)$ and then multiplying by 2. We now divide this quadrilateral into two isosceles right triangles with hypotenuse r and a rectangle with sides $\sqrt{2}N - 2r/\sqrt{2}$ and $r/\sqrt{2}$, obtaining that

$$\frac{1}{2}\text{area}(H_1(r)) = \frac{r^2}{2} + \frac{r}{\sqrt{2}}\left(\sqrt{2}N - \sqrt{2}r\right) = (2N - r)\frac{r}{2}. \tag{1}$$

More generally, if $r_2 > r_1$, we can compute area$(H_1(r_2) \setminus H_1(r_1))$ by computing area$(H_1(r_2))$ − area$(H_1(r_1))$. Thus,

$$\frac{1}{2}\text{area}(H_1(r_2) \setminus H_1(r_1)) = (2N - r_2)\frac{r_2}{2} - (2N - r_1)\frac{r_1}{2} = (2N - r_2 - r_1)\frac{r_2 - r_1}{2}. \tag{2}$$

We now use Equations (1) and (2) to compute the area of $H_{1,j}$ as follows:

$$\begin{aligned}\text{area}(H_{1,j}) &= \text{area}\left(H\left(C_j - \sqrt{2}\cdot 2^{j-3}\right)\right) + \text{area}\left(H(C_j)\setminus H\left(C_j - \sqrt{2}\cdot 2^{j-3}\right)\right)\\ &= \left(2N - C_j + \sqrt{2}\cdot 2^{j-3}\right)\left(C_j - \sqrt{2}\cdot 2^{j-3}\right) + \sqrt{2}\cdot 2^{j-3}\left(2N - 2C_j + \sqrt{2}\cdot 2^{j-3}\right)\\ &= \left(N + \sqrt{(\sqrt{2}N - 2^j)^2 - N^2}\right)\left(N - \sqrt{(\sqrt{2}N - 2^j)^2 - N^2}\right)\\ &\quad +\sqrt{2}\cdot 2^{j-3}\left(2\sqrt{(\sqrt{2}N - 2^j)^2 - N^2} - \sqrt{2}\cdot 2^{j-3}\right)\\ &\leq \left(N + \sqrt{(\sqrt{2}N - 2^j)^2 - N^2}\right)\left(N - \sqrt{(\sqrt{2}N - 2^j)^2 - N^2}\right)\\ &\quad + \sqrt{2}\cdot 2^{j-2}\sqrt{(\sqrt{2}N - 2^j)^2 - N^2}. \end{aligned} \tag{3}$$

Note that $(\sqrt{2}N - 2^j)^2 - (\sqrt{2}N - 2^{j-1})^2 = (\sqrt{2}N - 2^j + \sqrt{2}N - 2^{j-1})(2^{j-1} - 2^j) \leq 0$. Hence, $(\sqrt{2}N - 2^j)^2 - N^2 \leq (\sqrt{2}N - 2^{j-1})^2$. Using this observation on (3), we obtain

$$\begin{aligned}\text{area}(H_{1,j}) &\leq \left(N + \sqrt{(\sqrt{2}N - 2^j)^2 - N^2}\right)\left(N - \sqrt{(\sqrt{2}N - 2^j)^2 - N^2}\right) + \sqrt{2}\cdot 2^{j-2}\left(\sqrt{2}N - 2^{j-1}\right)\\ &= 2N^2 - \left(\sqrt{2}N - 2^j\right)^2 + \sqrt{2}\cdot 2^{j-2}\left(\sqrt{2}N - 2^{j-1}\right)\\ &= 2^{j+1}\left(\sqrt{2}N - 2^{j-1}\right) + \sqrt{2}\cdot 2^{j-2}\left(\sqrt{2}N - 2^{j-1}\right)\\ &= \left(1 + \frac{1}{8\sqrt{2}}\right)2^4 2^{j-3}\left(\sqrt{2}N - 2^{j-1}\right) \qquad (4)\\ &= \left(1 + \frac{1}{8\sqrt{2}}\right)2^4 \text{area}\left(R_j\right). \end{aligned}$$

This concludes the proof.

2.3 Hard Polygons

Recall that by Lemma 10, for only $O(\log N)$ classes $\mathcal{P}_j$ it holds that $\mathcal{P}_j \cap \mathcal{P}_H \neq \emptyset$. We first show that for each class $\mathcal{P}_j$, there are at most a constant number of polygons from $\mathcal{P}_j \cap \mathcal{P}_H$ in OPT (Figure 4).

Lemma 19. $|\mathcal{P}_j \cap \mathcal{P}_H \cap \text{OPT}| = O(1)$.

Proof. Recall that $\mathcal{P}_j := \{i \in \mathcal{P} \mid \ell_i \in [\sqrt{2}N - 2^j, \sqrt{2}N - 2^{j-1})\}$. For $i = 1, 2$, we define $\mathcal{D}_{i,j}$ as the polygons in $\mathcal{P}_j$ such that their diagonal is contained in $H_i(r_i)$ in OPT. By Claim 1, we have that $\mathcal{P}_j = \mathcal{D}_{1,j} \cup \mathcal{D}_{2,j}$. Hence, it is sufficient to show that $|\mathcal{D}_{i,j} \cap \mathcal{P}_j \cap \mathcal{P}_H \cap \text{OPT}| = O(1)$ for $i = 1, 2$. We show the case $i = 1$, as the case $i = 2$ follows from symmetric arguments.

We prove the lemma by partitioning the polygons into two sets. Define $\mathcal{F} \subseteq \mathcal{P}_j \cap \mathcal{P}_H \cap \text{OPT}$ as the polygons that fit completely into $H_1(C_j)$ as defined in Claim 1, and $\overline{\mathcal{F}}$ as $(\mathcal{P}_j \cap \mathcal{P}_H \cap \text{OPT}) \setminus \mathcal{F}$. We begin with the polygons in $\mathcal{F}$. Recall that by Lemma 11, we know that $\text{area}(P_i) \geq \frac{1}{2}\ell_i h_i$. Using that $P_i \in \mathcal{P}_j \cap \mathcal{P}_H$, we obtain that

$$\text{area}(P_i) \geq \frac{1}{2}\ell_i h_i > \frac{1}{16}\left(\sqrt{2}N - 2^j\right)\left(\sqrt{2}N - \ell_i\right) \geq \frac{1}{16}\left(\sqrt{2}N - 2^j\right)2^{j-1}.$$

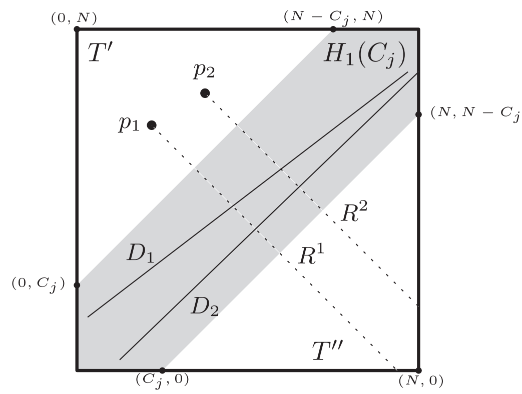


Fig. 4. Some of the elements that appear in the proof of Lemma 19.

Therefore,

$$\text{area}(\mathcal{F}) \geq \frac{1}{8}|\mathcal{F}|\left(\sqrt{2}N - 2^j\right)2^{j-2}.$$

Additionally, since all polygons in $\mathcal{F}$ completely fit into $H_1(C_j)$, we have that

$$\text{area}(\mathcal{F}) \leq \text{area}(H_1(C_j)) \leq 2^{j+2}(\sqrt{2}N - 2^{j-1}),$$

by using (4). We now combine these two facts and recall that, by Lemma 10, $j \leq \log\left((\sqrt{2}-1)N\right)+1$ to obtain

$$|\mathcal{F}| \leq 2^7\left(\frac{\sqrt{2}N - 2^{j-1}}{\sqrt{2}N - 2^j}\right) \leq 2^7\left(\frac{\sqrt{2}N}{\sqrt{2}N - 2^j}\right) \leq 2^7\left(\frac{\sqrt{2}N}{\sqrt{2}N - 2(\sqrt{2}-1)N}\right) \leq 2^9.$$

We now deal with $\overline{\mathcal{F}}$ (see Figure 4 for an illustration). Note that $K \setminus H_1(C_j)$ has two connected components: the triangle T' with vertices $(0,N)$, $(N - C_j, N)$, $(0, C_j)$ and T'' with vertices $(N, 0)$, $(N, N - C_j)$ and $(C_j, 0)$. We want to show that for each of these triangles, there is at most one polygon intersecting it.

Suppose, for the sake of contradiction, that there exist points $p_1, p_2 \in T'$ belonging to some polygons P_1 and P_2, respectively. Let $D^1 = u^1v^1$ and $D^2 = u^2v^2$ be diametrical segments of P_1 and P_2, respectively. Let r_1 and r_2 be as in Claim 1, and define $s = \max(r_1, r_2)$. By the same claim, we get that w.l.o.g. $\{u^1, v^1, u^2, v^2\} \subseteq H_1(s)$. We now consider the rays for $i = 1, 2$; $R_i := \left\{p_i + (\lambda, -\lambda) \mid \lambda \geq 0\right\}$. Note that since $s \leq C_j$, both R_1 and R_2 must intersect D^1 and D^2. Note that R_1 intersects first D^1 and then D^2 (otherwise, P_1 and P_2 would be intersecting) and R_2 intersects first D^2 and then D^1. Therefore, R_1 and R_2 intersect D^1 and D^2 in different order, which means that D^1 and D^2 must intersect, a contradiction. □

We describe now a quasi-polynomial time algorithm for hard polygons—that is, we want to prove Lemma 8. Lemmas 10 and 19 imply that $|\mathcal{P}_H \cap \text{OPT}| = O(\log N)$. Therefore, we can enumerate all possibilities for $\mathcal{P}_H \cap \text{OPT}$ in time $n^{O(\log N)}$. For each enumerated set $\mathcal{P}'_H \subseteq \mathcal{P}_H$, we need to check whether it fits into K. We cannot try all possibilities for placing $\mathcal{P}'_H$ into K since we are allowed to rotate the polygons in $\mathcal{P}'_H$ by arbitrary angles. To this end, we show that there is a subset of $\mathcal{P}_H \cap \text{OPT}$ of large weight which contains only a single polygon or which does not use the complete space of the knapsack but leaves some empty space. We use this empty space to move the polygons slightly and rotate them such that each of them is placed in one out of $(nN)^{O(1)}$ different positions that we can compute beforehand. Hence, we can guess all positions of these polygons in time $(nN)^{O(\log N)}$. We define that a *placement* of a polygon $P_i \in \mathcal{P}$ inside K is a polygon $\tilde{P}_i$ such that $d + \text{rot}_\alpha(P_i) = \tilde{P}_i \subseteq K$, where $d \in \mathbb{R}^2$ and $\text{rot}_\alpha(P_i)$ is the polygon that we obtain when we rotate P_i by an angle α clockwise around its first vertex $(x_{i,1}, y_{i,1})$.

Lemma 20. *For each polygon $P_i \in \mathcal{P}_H$, we can compute a set of $(nN)^{O(1)}$ possible placements $\mathcal{L}_i$ in time $(nN)^{O(1)}$ such that there exists a set $\mathcal{P}'_H \subseteq \mathcal{P}_H \cap \text{OPT}$ with $w(\mathcal{P}_H \cap \text{OPT})=O(w(\mathcal{P}'_H))$ which can be packed into K such that each polygon P_i is packed according to a placement in $\mathcal{L}_i$.*

Before proving Lemma 20, we observe that the number of classes $\mathcal{P}_j$ containing polygons whose respective values ℓ_i are at most $(\sqrt{2}-\epsilon)N$ is $O_\epsilon(1)$ for any given $\epsilon > 0$; recall that $\sqrt{2}N$ is the length of the diagonal of K. We state this observation, and similar ones, in the following lemma. Recall that $r_i = N - \sqrt{\ell_i^2 - N^2}$ for each polygon P_i (see Claim 1).

Lemma 21. *For each $\epsilon > 0$, there is a constant $k_\epsilon \in \mathbb{N}$ such that each polygon $P_i \in \bigcup_{j=j_{\min}}^{j_{\max}-k_\epsilon} \mathcal{P}_j$ and its placement satisfies that*

(1) $\ell_i \geq (\sqrt{2} - \epsilon)N$,
(2) $r_i \leq \epsilon N$.

Let v^1 and v^2 be the points which define the diametrical segment and recall the definitions of $T_1(r)$ and $T_2(r)$ as in Claim 1, then

(3) either $\{v^1, v^2\} \subseteq T_1(\epsilon N)$ or $\{v^1, v^2\} \subseteq T_2(\epsilon N)$.

Let α be the smallest angle between the diametrical segment and the bottom edge of the knapsack.

(4) $\pi/4 - \epsilon \leq \alpha \leq \pi/4 + \epsilon$,
(5) $\frac{\sqrt{2}}{2} - \epsilon \leq \sin\alpha \leq \frac{\sqrt{2}}{2} + \epsilon$,
(6) $\frac{\sqrt{2}}{2} - \epsilon \leq \cos\alpha \leq \frac{\sqrt{2}}{2} + \epsilon$.

We remark that we will use Lemma 21 only for very small values of ϵ. Then, ℓ_i is close to $\sqrt{2}N$ and r_i is close to 0.

Proof. Note that (3) is a direct consequence of (2) and Claim 1; (4) is a direct consequence of (3); and (5) and (6) are direct consequences of (4). Consequently, we only prove (1) and (2). Note that by choosing $k_\epsilon = j_{\max} - \log(\epsilon N)$, we get that

$$\ell_i \geq \sqrt{2}N - 2^{j_{\max}-k_\epsilon} = \left(\sqrt{2} - \epsilon\right) N.$$

We now prove (2). Let $\epsilon' = \frac{\epsilon}{2\sqrt{2}}$. By (1), we assume that $\ell_i \geq (\sqrt{2} - \epsilon')N$. Therefore,

$$\begin{aligned} r_i &\leq \left(1 - \sqrt{(\sqrt{2} - \epsilon')^2 - 1}\right) N \\ &\leq \left(1 - \sqrt{1 - 2\sqrt{2}\epsilon'}\right) N \\ &\leq 2\sqrt{2}\epsilon' N = \epsilon N. \end{aligned}$$

□

We now are ready to prove Lemma 20.

Proof of Lemma 20. Let $\epsilon > 0$ be a sufficiently small constant. Recall that $j_{\min} = -\lceil \log N \rceil - 1$ and $j_{\max} = 1 + \lfloor \log((\sqrt{2} - 1)N) \rfloor$, as in Lemma 10. Due to Lemmas 10 and 19, there can be only $O_\epsilon(1)$ hard polygons in $\text{OPT} \cap \bigcup_{j=j_{\max}-k_\epsilon+1}^{j_{\max}} \mathcal{P}_j$. Since these are constantly many, we can output the better among the following two solutions: the first solution consists of only one polygon which is the polygon in OPT $\cap \bigcup_{j=j_{\max}-k_\epsilon+1}^{j_{\max}} \mathcal{P}_j$ of highest weight, and the second solution is an $O(1)$-approximate solution for the hard polygons in $\text{OPT}' := \text{OPT} \cap \bigcup_{j=j_{\min}}^{j_{\max}-k_\epsilon} \mathcal{P}_j$. This will yield a constant approximation algorithm for hard polygons. Thus, it suffices to prove Lemma 20 for the hard polygons in $\text{OPT}' := \text{OPT} \cap \bigcup_{j=j_{\min}}^{j_{\max}-k_\epsilon} \mathcal{P}_j$. Since for each polygon $P_i \in \text{OPT}'$ it holds that $\ell_i \geq (\sqrt{2}-\epsilon)N$, we have that in any placement of P_i inside K the diametrical segment has to have essentially a $\pi/4$ angle between the diametrical segment with the edges of the knapsack. Furthermore, by losing a factor of 2, we can assume that the diametrical segment is oriented from the top-left corner to the bottom-right corner. Here and in the following, "by losing a factor of α" for some $\alpha > 1$ means that for a considered set (e.g., the optimal solution) there exists a subset with the claimed property, whose weight is by a factor of at most α smaller than the weight of the first set.

Let L denote the line segment connecting $p_L := (0, N/2)$, and $p_R := (N, N/2)$. Note that if ϵ is sufficiently small, then every polygon in OPT′ intersects L. Let L_1 denote the line segment connecting $p_M := (N/2, N/2)$ and p_R, and let L_2 denote the line segment connecting p_L and p_M (see Figure 6). Since every polygon intersecting L_1 and L_2 must also intersect the midpoint of the knapsack, OPT′ can only contain one such polygon. Thus, by losing a factor 3, we can assume that each polygon in OPT′ intersects L_1 but not L_2. We group the polygons in OPT′ into three groups: $\text{OPT}^{(1)}, \text{OPT}^{(2)}, \text{OPT}^{(3)}$. We define that $\text{OPT}^{(1)}$ contains the polygons in OPT′ that have empty intersection with $[0, \frac{1}{nN}] \times [0, N]$. We define that $\text{OPT}^{(2)}$ contains the polygons in $\text{OPT}' \setminus \text{OPT}^{(1)}$ that have empty intersection with $[0, N] \times [0, \frac{1}{nN}]$. Finally, we define $\text{OPT}^{(3)} := \text{OPT}' \setminus (\text{OPT}^{(1)} \cup \text{OPT}^{(2)})$.

Consider the group $\text{OPT}^{(1)}$. We sort the polygons in OPT′ in the order in which they intersect L from left to right; let $\text{OPT}' = \{Q_1, \ldots, Q_k\}$ denote this ordering. For each $i \in \{1, \ldots, k\}$, we translate each polygon Q_i to the left by $\frac{n-i+1}{n^2 N}$ units. We argue that between any two consecutive polygons Q_i, Q_{i+1}, there is some empty space that intuitively we can use as slack. Since Q_i, Q_{i+1} are convex, for their original placement there is a line L' that separates them. If ϵ is sufficiently small, then the line segments defining ℓ_i and ℓ_i' have essentially a $\pi/4$ angle with the edges of the knapsack. Since $\ell_i \geq (\sqrt{2}-\epsilon)N$ and $\ell_i' \geq (\sqrt{2}-\epsilon)N$, this implies that also L' essentially forms a $\pi/4$ angle with the edges of the knapsack. After translating Q_i and Q_{i+1}, we can draw not only a line separating them (like L') but instead a strip separating them, defined via two lines L'', L''' whose angle is identical to the angle of L', and such that the distance between L'' and L''' is at least $\Omega(\frac{1}{n^2 N})$. Next, we rotate Q_i around one of its vertices until the angle of the diametrical segment is a multiple of $\eta \frac{1}{n^2 N}$ for some small constant $\eta > 0$ to be defined later, or one of the vertices of Q_i touches an edge of the knapsack. In the latter case, let v be a vertex of Q_i that touches an edge of the knapsack. We rotate Q_i around v until the angle of the diametrical segment is a multiple of $\eta \frac{1}{n^2 N}$ or another vertex v' of Q_i touches an edge of the knapsack. In the latter case, we observe that two vertices of Q_i touch an edge of the knapsack. Since Q_i has at most $O(N)$ vertices, there are at most $O(N^2)$ such orientations for Q_i. Otherwise, there are only $n^2 N/\eta$ possibilities for the angle of the diametrical segment which gives at most $O(n^2 N/\eta)$ possible orientations for Q_i in total. Finally, we move Q_i to the closest placement with the property that the first vertex v of Q_i is placed on a position in which both coordinates are multiples of $\eta \frac{1}{n^2 N}$. One can show that due to the empty space between any two consecutive polygons Q_i, Q_{i+1}, no two polygons overlap after the movement, if η is chosen sufficiently small. This yields a placement for the polygons in $\text{OPT}^{(1)}$ in which each polygon Q_i is placed according to one out of $(nN)^{O(1)}$ positions.

We use a symmetric argumentation for the polygons in $\text{OPT}^{(2)}$. Finally, we want to argue that $|\text{OPT}^{(3)}| = O(1)$. Observe that each polygon $P_i \in \text{OPT}^{(3)}$ intersects L_1, the strip $[0, \frac{1}{nN}] \times [0, N]$, and the strip $[0, N] \times [0, \frac{1}{nN}]$, but has empty intersection with L_2. Since the diametrical segment is oriented from the top-left corner to the bottom-right corner, we conclude that the diametrical segment must have length at least $\sqrt{2}N - \frac{2\sqrt{2}}{nN}$. Therefore, $P_i \in \mathcal{P}_j$ with $2^j \leq \frac{2\sqrt{2}}{nN}$ and then $j \leq \log(2\sqrt{2}) - \log(nN)$. First assume that $P_i \in \mathcal{P}_j$. If n is a sufficiently large constant, we have that $j < j_{\min}$ which contradicts Lemma 10. However, if $n = O(1)$, then the claim is trivially true since then $|\text{OPT}| = O(1)$. □

This yields the proof of Lemma 8.

2.4 Hard Triangles

In this section, we present an $O(1)$-approximation algorithm in *polynomial* time for hard polygons assuming that they are all triangles (i.e., we prove Lemma 9). Slightly abusing notation, denote by OPT the set $\mathcal{P}'_H$ obtained by applying Lemma 20. We distinguish the triangles in OPT into two types: *edge-facing* triangles and *corner-facing* triangles. Let $P_i \in \text{OPT} \cap \mathcal{P}_H$, let e_1, e_2 denote the two longest edges of P_i, and let v_i^* denote the vertex of P_i adjacent to e_1 and e_2. Let $R_i^{(1)}$ and $R_i^{(2)}$ be the two rays that originate at v_i^* and that contain e_1 and e_2, respectively, in the placement of P_i in OPT. We have that $R_i^{(1)} \setminus \{v_i^*\}$ and $R_i^{(2)} \setminus \{v_i^*\}$ intersect at most one edge of the knapsack each. If $R_i^{(1)} \setminus \{v_i^*\}$ and $R_i^{(2)} \setminus \{v_i^*\}$ intersect the same edge of the knapsack, then we say that P_i is *edge-facing, and* if one of them intersects a horizontal edge and the other one intersects a vertical edge, we say that P_i is *corner-facing*. The next lemma shows that there can be only $O(1)$ triangles in $\text{OPT} \cap \mathcal{P}_H$ that are neither edge- nor corner-facing, and therefore we compute an $O(1)$-approximation with respect to the total profit of such triangles by simply selecting the input triangle with maximum weight.

Lemma 22. *The number of triangles in* $\text{OPT} \cap \mathcal{P}_H$ *that are neither edge-facing nor corner-facing is* $O(1)$.

Proof. Let $P_i \in \text{OPT} \cap \mathcal{P}_H$ that is neither edge-facing nor corner-facing. Assume w.l.o.g. that both $R_i^{(1)} \setminus \{v_i^*\}$ and $R_i^{(2)} \setminus \{v_i^*\}$ intersect a horizontal edge of the knapsack. Let e_1, e_2 denote the two longest edges of P_i. Since P_i is hard, we know that one of these edges is longer than N and therefore the other one is longer than $N/2$. Let α denote the angle between e_1 and e_2. It holds that α cannot be arbitrarily small since otherwise it cannot be that $R_i^{(1)} \setminus \{v_i^*\}$ and $R_i^{(2)} \setminus \{v_i^*\}$ intersect different horizontal edges of the knapsack. Formally, assume w.l.o.g. that v_i^* lies in the upper half of the knapsack and that $R_i^{(1)} \setminus \{v_i^*\}$ intersects the bottom edge of the knapsack. Let β denote the angle between the horizontal line and the line going through v_i^* and $R_i^{(1)}$ (note that $\alpha \geq \beta$). Let A be the intersection of $R_i^{(1)}$ and the bottom edge of the knapsack and B the projection of v_i^* onto the bottom edge of the knapsack (i.e., $B = ((v_i^*)_x, 0)$). Note that $\tan \beta$ is the quotient between the length of the line segment $\overline{v_i^* B}$ and the length of the line segment $\overline{AB}$, which implies that $\tan \beta \geq \frac{N}{2N} = 1/2$. Consequently, $\alpha \geq \beta \geq \arctan(1/2) \geq 0.4$. Therefore, $\text{area}(P_i) \geq \Omega(N^2)$. Hence, there can be at most $O(1)$ triangles in $P_i \in \text{OPT} \cap \mathcal{P}_H$ that are neither edge-facing nor corner-facing. □

Let p_{TL}, p_{TR}, p_{BL}, and p_{BR} denote the top left, top right, bottom left, and bottom right corners of K, respectively, and let $p_M := (N/2, N/2)$, $p_L := (0, N/2)$, and $p_R := (N, N/2)$ (see Figure 6). By losing a factor $O(1)$, we assume from now on that OPT contains at most one hard triangle from each group $\mathcal{P}_j$, using Lemma 19.

Let $\text{OPT}_{\text{EF}} \subseteq \text{OPT} \cap \mathcal{P}_H$ denote the edge-facing hard triangles in OPT and denote by $\text{OPT}_{\text{CF}} \subseteq \text{OPT} \cap \mathcal{P}_H$ the corner-facing hard triangles in OPT. In the remainder of this section, we present an

$O(1)$-approximation algorithms for edge-facing and for corner-facing triangles in $\mathcal{P}_H$. By selecting the best solution among the two, we obtain the proof of Lemma 9.

2.4.1 Edge-Facing Triangles. We define a special type of solutions called *top-left-packings* that our algorithm will compute. We will show later that there are solutions of this type whose profit is at least a constant fraction of the profit of $\mathrm{OPT}_{\mathrm{EF}}$.

For each $t \in \mathbb{N}$, let $p_t := p_M + (\frac{t}{N^2}, 0)$. Let $\mathcal{P}' = \{P_{i_1}, \ldots, P_{i_k}\}$ be a set of triangles that are ordered according to the groups $\mathcal{P}_j$—that is, such that for any $P_{i_\ell}, P_{i_{\ell+1}} \in \mathcal{P}'$ with $P_{i_\ell} \in \mathcal{P}_j$ and $P_{i_{\ell+1}} \in \mathcal{P}_{j'}$ for some j, j' it holds that $j \leq j'$. We define a placement of $\mathcal{P}'$ that we call a *top-left-packing*. First, we place P_{i_1} such that $v^*_{i_1}$ coincides with p_{TL} and one edge of P_{i_1} lies on the diagonal of K that connects p_{TL} and p_0. Note that there is a unique way to place P_{i_1} in this way. Iteratively, suppose that we have packed triangles $\{P_{i_1}, \ldots, P_{i_\ell}\}$ such that for each triangle $P_{i_{\ell'}}$ in this set its respective vertex $v^*_{i_{\ell'}}$ coincides with p_{TL} (see Figure 1(b)). Intuitively, we pack $P_{i_{\ell+1}}$ on top of P_{i_ℓ} such that $v^*_{i_{\ell+1}}$ coincides with p_{TL}. Let t be the smallest integer such that the line segment connecting p_t and p_R has empty intersection with each triangle $P_{i_1}, \ldots, P_{i_\ell}$ according to our placement. We place $P_{i_{\ell+1}}$ such that $v^*_{i_{\ell+1}}$ coincides with p_{TL} and one of its edges lies on the line that contains p_{TL} and p_t. There is a unique way to place $P_{i_{\ell+1}}$ in this way. We continue until we placed all triangles in $\mathcal{P}'$. If all of them are placed completely inside K, we say that the resulting solution is a *top-left-packing* and that $\mathcal{P}'$ is *top-left-packable.* We define *bottom-right-packing* and *bottom-right-packable* symmetrically, mirroring the preceding definition along the line that contains p_{BL} and p_{TR}. Thus, in such a packing, each packed triangle P is contained in the triangular region defined by the vertices p_{TL}, p_{TR}, and p_{BR} and one of the vertices of P is placed at p_{BR}.

In the next lemma, we show that there is always a top-left-packable or a bottom-right-packable solution with large profit compared to $\mathcal{P}_H \cap \mathrm{OPT}$ or there is a single triangle with large profit.

Lemma 23. *There exists a solution $\mathcal{P}^*_H \subseteq \mathcal{P}_H \cap \mathrm{OPT}_{\mathrm{EF}}$ such that $w(\mathcal{P}_H \cap \mathrm{OPT}_{\mathrm{EF}})=O(w(\mathcal{P}^*_H))$ and*

- *$\mathcal{P}^*_H$ is top-left-packable or bottom-right-packable and for each j we have that $|\mathcal{P}^*_H \cap \mathcal{P}_j| \leq 1$,*
- *or it holds that $|\mathcal{P}^*_H| = 1$.*

We will prove Lemma 23 later in Section 2.4.2. We describe now a polynomial time algorithm that computes the most profitable solution that satisfies the properties of Lemma 23. Assuming Lemma 23, this yields an $O(1)$-approximation with respect to the edge-facing triangles in OPT.

To find the most profitable solution $\mathcal{P}^*_H$ that satisfies $|\mathcal{P}^*_H| = 1$, we simply take the triangle with maximum weight. Let P_{i^*} be this triangle. We establish now a DP that computes the most profitable top-left-packable solution; computing the most profitable bottom-right-packable solution works analogously. Our DP has a cell corresponding to pairs (j, t) with $j, t \in \mathbb{Z}$. Intuitively, (j, t) represents the subproblem of computing a set $\mathcal{P}'_H \subseteq \mathcal{P}_H$ of maximum weight such that $\mathcal{P}'_H \cap \mathcal{P}_{j'} = \emptyset$ for each $j' < j$ and $|\mathcal{P}'_H \cap \mathcal{P}_{j''}| \leq 1$ for each $j'' \geq j$ and such that $\mathcal{P}'_H$ is top-left-packable inside the triangular area T_t defined by the line that contains p_{TL} and p_t, the top edge of K, and the right edge of K. Given a cell (j, t) we want to compute a solution $DP(j, t)$ associated with (j, t). Intuitively, we guess whether the optimal solution $\mathcal{P}'_H$ to (j, t) contains a triangle from $\mathcal{P}_H \cap \mathcal{P}_j$. Therefore, we try each triangle $P_i \in \mathcal{P}_H \cap \mathcal{P}_j$ and place it inside T_t such that v^*_i coincides with p_{TL} and one of its edges lies on the line containing p_{TL} and p_t. Let $t'(P_i)$ denote the smallest integer such that $t'(P_i) \geq t$ and $p_{t'(P_i)}$ is not contained in the resulting placement of P_i inside T_t. We associate with P_i the solution $P_i \cup DP(j + 1, t'(P_i))$. Finally, we define $DP(j, t)$ to be the solution of maximum profit among the solutions $P_i \cup DP(j + 1, t'(P_i))$ for each $P_i \in \mathcal{P}_H \cap \mathcal{P}_j$ and the solution $DP(j + 1, t)$.

We introduce a DP-cell $DP(j, t)$ for each pair $(j, t) \in \mathbb{Z}^2$ where $j_{\min} \leq j \leq j_{\max}$ and $0 \leq t \leq \log_{1+1/n}\left(\frac{N}{2}\right)$. Note that due to Lemma 10, for all other values of j we have that $\mathcal{P}_j \cap \mathcal{P}_H = \emptyset$. Also note that $p_t \notin K$ if $t \geq N^2/2$. This yields at most $(nN)^{O(1)}$ cells in total. Finally, we output the solution $DP(j_{\min}, 0)$.

In the next lemma, we prove that our DP computes the optimal top-left-packable solution with the properties of Lemma 23.

Lemma 24. *There is an algorithm with a running time of $(nN)^{O(1)}$ that computes the optimal solution $\mathcal{P}' \subseteq \mathcal{P}_H$ such that $\mathcal{P}'$ is top-left-packable or bottom-right-packable and such that for each j we have that $|\mathcal{P}' \cap \mathcal{P}_j| \leq 1$.*

Proof. We say that a set of triangles S is a (j,t)-solution if it is top-left-packable inside of T_t, and only uses items from $\bigcup_{j'' \geq j} \mathcal{P}_H \cap \mathcal{P}_{j''}$ and $|\mathcal{P}_H \cap \mathcal{P}_{j''}| \leq 1$ for each $j'' \geq j$. Let $\text{OPT}_{j,t}$ be the (j,t)-solution of maximum weight. We aim to show that $\text{OPT}_{j,t} = DP(j,t)$ for each $j_{\min} \leq j \leq j_{\max}$ and $0 \leq t \leq \log_{1+\frac{1}{n}}\left(\frac{N}{2}\right)$.

We proceed by backward induction on j. If $j = j_{\max}$, then $\text{OPT}_{j,t}$ is exactly the packing of the top-left-packable triangle of maximum weight in T_t. Since $DP(j,t)$ tries to top-left-pack all triangles into T_t, it is clear that $\text{OPT}_{j,t} = DP(j,t)$.

We now deal with the case $j < j_{\max}$. By induction, $DP(j,t)$ is the solution of maximum profit among $P_i \cup \text{OPT}(j+1, t'(P_i))$ for $P_i \in \mathcal{P}_H \cap \mathcal{P}_j$ and $\text{OPT}(j+1,t)$. Suppose, by contradiction, that there exists a (j,t)-solution S such that $w(S) > w(DP(j,t))$. We consider two cases. If $|S \cap \mathcal{P}_j| \neq \emptyset$, we select $P_{i^*} \in S \cap \mathcal{P}_j$ and note that

$$w(P_{i^*}) + w(\text{OPT}(j+1, t'(P_{i^*}))) \leq w(DP(j,t)) < w(P_{i^*}) + w(S \setminus \{P_{i^*}\}).$$

Therefore, $w(S \setminus \{P_{i^*}\}) > w(\text{OPT}(j+1, t'(P_{i^*})))$, which contradicts the optimality of $w(\text{OPT}(j+1, t'(P_{i^*})))$, since they are both $(j+1, t'(P_{i^*}))$-solutions. For the case $|S \cap \mathcal{P}_j| = \emptyset$, we have

$$\text{OPT}(j+1, t) \leq w(DP(j,t)) < w(S).$$

Hence, $w(S) > w(\text{OPT}(j+1,t))$, which contradicts the optimality of $\text{OPT}(j+1,t)$, as they are both $(j+1,t)$-solutions.

We conclude that $\text{OPT}_{j,t} = DP(j,t)$ for each $j_{\min} \leq j \leq j_{\max}$ and $0 \leq t \leq \log_{1+\frac{1}{n}}\left(\frac{N}{2}\right)$. In particular, $DP(j_{\min}, 0) = \text{OPT}_{j,0}$, as desired. Note that the bottom-right-packable case can be dealt with in a similar manner, concluding the proof. □

We execute the preceding DP and its counterpart for bottom-right-packable solutions to obtain a top-left-packable solution $\mathcal{P}'_1$ and a bottom-right-packable solution $\mathcal{P}'_2$. We output the most profitable solution among $\{P_{i^*}\}, \mathcal{P}'_1, \mathcal{P}'_2$. Due to Lemma 23, this yields a solution with weight at least $\Omega(w(\mathcal{P}_H \cap \text{OPT}))$.

Lemma 25. *There is an algorithm with a running time of $(nN)^{O(1)}$ that computes a solution $\mathcal{P}'_H \subseteq \mathcal{P}_H$ such that $w(\text{OPT}_{\text{EF}}) = O(w(\mathcal{P}'_H))$.*

2.4.2 Existence of Profitable Top-Left- or Bottom-Right-Packable Solution. In this subsection, we prove Lemma 23. Let $\epsilon > 0$ be a constant to be defined later. Like in the proof of Lemma 20, we observe that there can be only $O_\epsilon(1)$ classes $\mathcal{P}_j$ containing polygons whose respective values ℓ_i are not larger than $(\sqrt{2} - \epsilon)N$; recall that $\sqrt{2}N$ is the length of the diagonal of K. Furthermore, for the polygons with diameter at least $(\sqrt{2} - \epsilon)N$, we have that in any placement of P_i inside K the angle α between the diametrical segment and the bottom edge of the knapsack is essentially $\pi/4$ by Lemma 21. Additionally, we have that $\sin\alpha$ and $\cos\alpha$ are essentially $\frac{\sqrt{2}}{2}$. Finally, recall that $r_i = N - \sqrt{\ell_i^2 - N^2}$ as defined in Lemma 17 is at most ϵN by Lemma 21.

Due to Lemmas 10 and 19, there can be only $O_\epsilon(1)$ hard polygons in $\mathrm{OPT} \cap \bigcup_{j=j_{\max}-k_\epsilon+1}^{j_{\max}} \mathcal{P}_j$. Hence, it suffices to prove the claim for the hard polygons in $\mathrm{OPT}_W := \mathrm{OPT}_{\mathrm{EF}} \cap \bigcup_{j=j_{\min}}^{j_{\max}-k_\epsilon} \mathcal{P}_j$ since otherwise the second case of Lemma 23 applies if we define that $\mathcal{P}_H^*$ contains the polygon in $\mathrm{OPT}_{\mathrm{EF}}$ of maximum weight. Note that it holds that each triangle $P_i \in \mathrm{OPT}_W$ intersects the line segment L that we define to be the line segment that connects p_L with p_R. Let L_1 denote the subsegment of L that connects p_M with p_R, and let L_2 denote the line segment connecting p_L with p_M. Now each triangle in OPT_W either overlaps p_M or intersects L_1 but not L_2 or it intersects L_2 but not L_1. Therefore, by losing a factor of 3, we can restrict ourselves to one of these cases.

Lemma 26. *If $\epsilon > 0$ is sufficiently small, then by losing a factor 3 we can assume that for each triangle $P_i \in \mathrm{OPT}_W$ we have that $P_i \cap L = P_i \cap L_1$ or that $|\mathrm{OPT}_W| = 1$.*

Proof. There can be at most one triangle $P_{i^*} \in \mathrm{OPT}_W$ that overlaps p_M. Each other triangle $P_i \in \mathrm{OPT}_W$ satisfies that $P_i \cap L = P_i \cap L_1$ or that $P_i \cap L = P_i \cap L_2$. If the triangles $P_i \in \mathrm{OPT}_W$ satisfying $P_i \cap L = P_i \cap L_1$ have a total weight of at least $\frac{1}{3}w(\mathrm{OPT}_W)$ or if $w_{i^*} \geq \frac{1}{3}w(\mathrm{OPT}_W)$, then we are done. Otherwise, the triangles satisfying that $P_i \cap L = P_i \cap L_2$ have a total weight of at least $\frac{1}{3}w(\mathrm{OPT}_W)$, and we establish the claim of the lemma by rotating OPT by π. □

If $|\mathrm{OPT}_W| = 1$, then we are done. Therefore, we assume that $P_i \cap L = P_i \cap L_1$ for each $P_i \in \mathrm{OPT}_W$. In the next lemma, we prove that by losing a factor of $O(1)$ we can assume that the triangles in OPT_W intersect L_1 in the order of their groups $\mathcal{P}_j$ (assuming that ϵ is a sufficiently small constant). We call such a solution *group-respecting* as defined next.

Definition. Let $\mathcal{P}' = \{P_{i_1}, \dots, P_{i_k}\}$ be a solution in which each triangle intersects L_1, and assume w.l.o.g. that the triangles in $\mathcal{P}'$ intersect L_1 in the order $P_{i_1}, \dots, P_{i_k}$ when going from p_M to p_R. We say that $\mathcal{P}'$ is *group-respecting* if for any two triangles $P_{i_\ell}, P_{i_{\ell+1}} \in \mathcal{P}'$ with $P_{i_\ell} \in \mathcal{P}_j$ and $P_{i_{\ell+1}} \in \mathcal{P}_{j'}$ for some j, j' it holds that $j \leq j'$.

For each $P_i \in \mathrm{OPT}_W$, let d_i denote the length of the intersection of P_i and L_1 in the placement of OPT.

Lemma 27. *If $\epsilon > 0$ is sufficiently small, then by losing a factor $O(1)$ we can assume that OPT_W is group-respecting and that $|\mathrm{OPT}_W \cap \mathcal{P}_j| \leq 1$ for each j.*

Proof. Due to Lemma 19, we lose only a factor $O(1)$ by requiring that $|\mathrm{OPT}_W \cap \mathcal{P}_j| \leq 1$ for each j. We prove now that by losing another factor $O(1)$, we can assume that OPT_W is group-respecting. Let $P_i \in \mathcal{P}_j \cap \mathcal{P}_H$. Let D be the longest edge of P_i in the placement of P_i in OPT_W. Let α be the angle between D and L_1.

Like in the proof of Lemma 17, we define $r_i := N - \sqrt{\ell_i^2 - N^2}$. Intuitively, using r_i, we can define two hexagonal areas $H_1(r_i), H_2(r_i)$ close to the diagonals of the knapsack such that the longest edge of P_i lies within $H_1(r_i)$ or within $H_2(r_i)$. Call r_i' the length of the line segment $L_1 \cap H_1(r_i)$. By a similarity argument, we have that $r_i' = r_i$.

Let $\tilde{B}_i$ denote the rectangle obtained by taking the bounding box B_i of P_i and moving and rotating it such that one of its edges coincides with the diametrical segment of P_i (in the placement of OPT) inside K. Note that the intersection of $\tilde{B}_i$ and L_1 has length at most $\frac{h_i}{\sin\alpha} \leq \frac{1}{1-\epsilon}h_i \leq 2h_i$ by Lemma 21. Therefore, $d_i \leq 2h_i$. However, if ϵ is sufficiently small, then $h_i = O(d_i)$ and thus $d_i = \Theta(h_i)$. Additionally, it holds that $r_i = \Theta(h_i')$ and hence $d_i = \Omega(r_i)$. Let γ be a constant such that $d_i \geq \gamma \cdot r_i$ for each $P_i \in \mathrm{OPT}_W$.

Consider the triangle T given by $(0,0)$, (N,N), and $(N - r_i, N)$. Let α' be the angle at $(0,0)$. Note that the angles at (N,N) and $(N - r_i, N)$ are $\pi/4$ and $3\pi/4 - \alpha'$. Let q be a point in the diagonal of the knapsack at distance ℓ_i from $(0,0)$. Note that this splits T into an isosceles triangle, and the triangle given by q, $(N-r_i, N)$, and (N,N). Applying Lemma 21 and the law of sines on the latter triangle, we obtain that

$$\sqrt{2}N - \ell_i \leq r_i = \frac{\sin(\pi/2 + \alpha'/2)}{\sin(\pi/2 - \alpha'/2)}(\sqrt{2}N - \ell_i) \leq \frac{\sin(\pi/2 + \pi/10)}{\sin(\pi/2 - \pi/10)}(\sqrt{2}N - \ell_i) \leq 3(\sqrt{2}N - \ell_i).$$

In particular, if $i \in \mathcal{P}_j$ and $i' \in \mathcal{P}_{j'}$ such that $j + \Gamma \leq j'$ for $\Gamma = O(1)$, a simple induction on Γ shows that $r_i \leq 3^\Gamma r_{i'}$. Choosing $\Gamma = \log_3 \gamma$, we get that $r'_i = r_i \leq \gamma r_{i'} = d_{i'}$.

Let D be the longest edge of $P_{i'}$. Assume w.l.o.g. that D lies within $H_1(r_i)$. However, we have that $r'_i \leq d_{i'}$. Since P_i and $P_{i'}$ do not intersect, they must therefore be placed in a group-respecting manner in OPT_W.

We split OPT_W into Γ groups such that for each offset $a \in \{0, \ldots, \Gamma-1\}$, we define $\mathrm{OPT}_W^{(a)} := \mathrm{OPT}_W \cap \bigcup_{k \in \mathbb{Z}} \mathcal{P}_{a+k\Gamma}$. Therefore, for each solution $\mathrm{OPT}_W^{(a)}$ it holds that for any two distinct polygons $P_i \in \mathrm{OPT}_W^{(a)} \cap P_j$, $P_{i'} \in \mathrm{OPT}_W^{(a)} \cap P_{j'}$ for values j, j' it holds that P_i and $P_{i'}$ are placed in a group-respecting manner in OPT_W. Then taking the most profitable solution among the solutions $\{\mathrm{OPT}_W^{(a)}\}_{a \in \{0,\ldots,\Gamma-1\}}$ loses at most another factor $\Gamma = O(1)$. □

For each triangle $P_i \in \mathrm{OPT}_W$, let v_i^* be the vertex adjacent to the two longest edges of P_i in the placement of P_i in OPT. Additionally, let θ_i denote the angle at v_i^*. We prove that there exist only constantly many triangles $P_i \in \mathrm{OPT}_W$ with $\theta_i > \epsilon$.

Lemma 28. *There exist at most $O_\epsilon(1)$ triangles in $P_i \in \mathrm{OPT}_W$ such that $\theta_i > \epsilon$.*

Proof. Let $P_i \in \mathrm{OPT}_W$. Recall that the longest edge of P_i is horizontal. We now split P_i by a vertical line which crosses the midpoint of the longest edge. Note that this has split P_i into two polygons. Furthermore, one of these polygons contains v_i^* and is a triangle. We denote this triangle by T. Thus,

$$\mathrm{area}(P_i) \geq \mathrm{area}(T) = \frac{\ell_i^2}{8}\tan\theta_i > \frac{\ell_i^2}{8}\tan\epsilon \geq \frac{(\sqrt{2}-\epsilon)^2}{8}N^2 \tan\epsilon.$$

Hence, there can only be at most $O_\epsilon(1)$ such triangles in OPT_W. □

There are only constantly many triangles P_i in OPT such that $\theta_i > \epsilon$. If they contribute a constant fraction of the profit of OPT (e.g., at least OPT/2), then we obtain an $O(1)$-approximate solution by simply guessing the most profitable such triangle and guessing a packing for it inside the knapsack (e.g., there is always a packing in which one of the vertices of the triangle is in a corner, we can guess this vertex, and then easily obtain a packing of the triangle inside of the knapsack). Therefore, by losing a factor of at most 2, we can assume that $\theta_i \leq \epsilon$ for each $P_i \in \mathrm{OPT}_W$. Furthermore, since each triangle $P_i \in \mathrm{OPT}_W$ is very wide, v_i^* must be close to one of the four corners of K since otherwise the longest edge of P_i does not fit into K. Thus, by losing a factor 4, we assume that v_i^* is close to p_{TL} for each $P_i \in \mathrm{OPT}_W$.

Lemma 29. *Let $\epsilon > 0$. By losing a factor 4, we can assume for each $P_i \in \mathrm{OPT}_W$ that $\left\| v_i^* - p_{TL} \right\|_2 \leq 2\epsilon N$.*

Proof. Let u, v be the vertices that define ℓ_i. By Claim 1, we know that u or v is at distance r_i of some corner v_C of K. W.l.o.g. and applying Lemma 21, we assume that $\|v - v_C\| \leq \epsilon N$. Recall that by Lemma 21, $\|u - v\| = \ell_i \geq (\sqrt{2} - \epsilon)N$. Thus,

$$\begin{aligned}\|u - v_C\| &= \|u - v - (v_C - v)\| \\ &\geq \|u - v\| - \|v_C - v\| \\ &\geq \left(\sqrt{2} - \epsilon\right) N - \epsilon N \\ &= \left(\sqrt{2} - 2\epsilon\right) N.\end{aligned}$$

Let $B(x, r) := \{p \mid \|x - p\| \leq r\}$, and call $v_{C'}$ the corner farthest away from v_C. Note that $u \in K \setminus B(v_C, (\sqrt{2} - 2\epsilon)N) \subseteq B(v_{C'}, 2\epsilon N)$, from which we conclude that every endpoint of the diagonal is at distance at most $2\epsilon N$ from a corner. In particular, v_i^* must be a distance at most $2\epsilon N$ from some corner.

We define $N_{TL} := \{P_i \in \text{OPT}_W | \|v_i^* - p_{TL}\|_2 \leq 2\epsilon N\}$ and N_{TL}, N_{BL}, N_{BR} in a similar fashion. These sets partition OPT_W into four sets. Note that one of these sets must have weight at least $\frac{1}{4} w(\text{OPT}_W)$. If this set is N_{TL}, we are done. Otherwise, we simply rotate OPT_W accordingly. □

Due to Lemma 29, if ϵ is sufficiently small, we have for each triangle $P_i \in \text{OPT}_W$ that both $R_i^{(1)} \setminus \{v_i^*\}$ and $R_i^{(2)} \setminus \{v_i^*\}$ intersect the right edge of the knapsack or both $R_i^{(1)} \setminus \{v_i^*\}$ and $R_i^{(2)} \setminus \{v_i^*\}$ intersect the bottom edge of the knapsack. We call triangles P_i of the former type *right-facing* triangles, and we call the triangles of the latter type *bottom-facing* triangles.

Proposition 30. *If ϵ is sufficiently small, we have that by losing a factor of 2 we can assume that each triangle in OPT_W is right-facing or bottom-facing.*

Assume that $\text{OPT}_W = \{P_{i_1}, \ldots, P_{i_{|\text{OPT}_W|}}\}$. We partition OPT_W into $g = O(1)$ groups such that each group is top-left-packable. Then the most profitable such group yields a g-approximation. We initialize $\text{OPT}_W^{(1)} := \text{OPT}_W^{(2)} := \cdots := \text{OPT}_W^{(g)} := \emptyset$ and $k := 0$. Suppose inductively that for some $k \in \mathbb{N}_0$, we partitioned the triangles $P_{i_1}, \ldots, P_{i_{k-1}}$ into $\text{OPT}_W^{(1)}, \ldots, \text{OPT}_W^{(g)}$ such that each of these sets is top-left-packable. We argue that there is one value $t \in \{1, \ldots, g\}$ such that $\text{OPT}_W^{(t)} \cup \{P_{i_k}\}$ is also top-left-packable. To this end, observe that in the top-left-packing of each set $\text{OPT}_W^{(t)}$, each triangle $P_i \in \text{OPT}_W^{(t)}$ blocks a certain portion of L_1 such that no other triangle in this packing can overlap this part of L_1. For each triangle $P_i \in \left\{P_{i_1}, \ldots, P_{i_{k-1}}\right\}$, let $t(i) \in \mathbb{N}_0$ be the smallest integer t such that if $P_i \in \text{OPT}_W^{(g')}$ for some $g' \in \{1, \ldots, g\}$, then in the top-left-packing of $\text{OPT}_W^{(g')}$ the longest edge e of P_i lies on the line that contains p_{TL} and p_t. Additionally, let $t'(i)$ be the smallest integer t' such that $t(i) < t'$ and P_i does not overlap the point $p_{t'}$. Then, after placing P_i, we cannot add another triangle in a top-left-packing to $\text{OPT}_W^{(t)}$ that touches the subsegment of L_1 that connects $p_{t(i)}$ with $p_{t'(i)}$. Hence, intuitively, P_i blocks the latter subsegment. We define $\hat{d}_i := \left\|p_{t'(i)} - p_{t(i)}\right\|_2$. Our crucial insight is that up to a constant factor, in our top-left-packing the triangle P_i blocks as much of L_1 as it covers of L_1 in OPT_W.

Lemma 31. *If ϵ is sufficiently small, then for each triangle $P_i \in \text{OPT}_W$ it holds that $\hat{d}_i = O(d_i)$.*

Proof. We argue in a similar way as in the proof of Lemma 27. Let D be the longest edge of P_i in an arbitrary placement of P_i inside K. Let $\tilde{d}_i$ denote the length of the intersection of P_i and L_1 in this placement. Let α be the angle between D and L_1. Due to Proposition 21, we can assume that α satisfies $\pi/4 - 1/10 \leq \alpha \leq \pi/4 + 1/10$. Hence, if ϵ is a sufficiently small, then the intersection of $\tilde{B}_i$ and L_1 has length at most $2h_i$. Therefore, $\tilde{d}_i \leq 2h_i$.

However, if ϵ sufficiently small, then $h_i = O(\tilde{d}_i)$. Hence, $\tilde{d}_i = \Theta(h_i)$ and also $d_i = \Theta(h_i)$ and therefore $\tilde{d}_i = \Theta(d_i)$. Since $h_i \geq h_i'/8$, this implies that $d_i = \Omega(h_i')$. Additionally, it holds that

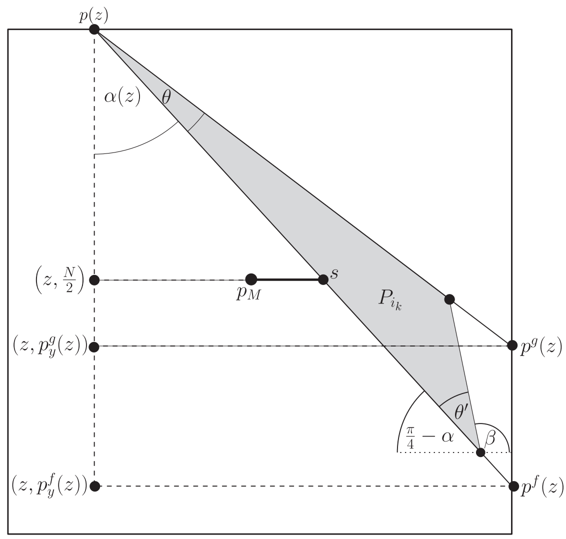


Fig. 5. The points, line segments, and angles used in the proof of Lemma 32.

$r_i = \Theta(h'_i)$ and D lies in $H_1(r_i)$ or $H_2(r_i)$. Furthermore, if n' is the number of packed triangles, we have that $\tilde{d}_i \leq \hat{d}_i + \frac{n'}{N^2}$ as each packed triangle contributes at most $1/N^2$ error. Since we pack at most N triangles, we obtain that $\hat{d}_i = O(\tilde{d}_i) = O(d_i)$. □

Lemma 31 implies that if g is a sufficiently large constant, then there is a value $t \in \{1, \ldots, g\}$ such that $\sum_{P_{i_\ell} \in \mathrm{OPT}_W^{(t)}} \hat{d}_{i_\ell} \leq \sum_{P_{i_\ell} \in \{P_{i_1}, \ldots, P_{i_{k-1}}\}} d_{i_\ell}$. Hence, in the top-left-packing for $\mathrm{OPT}_W^{(t)}$ (which at this point contains only triangles from $\{P_{i_1}, \ldots, P_{i_{k-1}}\}$), the triangles block less of L_1 than the amount of L_1 that the triangles $P_{i_1}, \ldots, P_{i_{k-1}}$ cover in OPT_W. However, we know that in OPT_W, the triangle P_{i_k} is placed such that it intersects L_1 farther on the right than any triangle in $P_{i_1}, \ldots, P_{i_{k-1}}$ due to Lemma 27. Using this, in the next lemmas we show that we can add P_{i_k} to $\mathrm{OPT}_W^{(t)}$.

Lemma 32. *If $\epsilon > 0$ is sufficiently small and if each triangle in OPT_W is right-facing, then we have that $\mathrm{OPT}_W^{(t)} \cup \{P_{i_k}\}$ is top-left-packable.*

Proof. Let $s^* = \sum_{P_{i_\ell} \in \mathrm{OPT}_W^{(t)}} \hat{d}_{i_\ell}$ and $s = (s^* + N/2, N/2)$. Consider $p(z) = (z, N)$, and define the point $p^f(z) = (p_x^f(z), p_y^f(z))$ as the intersection between the right side of the knapsack and $\overline{p(z)s}$ (see Figure 5 for an illustration). Here, the role of z is to parameterize the position of the "top-left corner" of the triangle P_{i_k}. Similarly, define $p^g(z) = (p_x^g(z), p_y^g(z))$ as the intersection between $\{N\} \times \mathbb{R}$ and the line L^θ obtained by rotating $\overline{p(z)s}$ around $p(z)$ by $\theta := \theta_{i_k}$ counterclockwise. Note that every polygon in $\mathrm{OPT}_W^{(t)}$ is contained in $H^- := K \cap \mathrm{conv}(\{p_{TL}, p^f(0), p_{BR}, p_{BL}\})$ as they are top-left-packed.

Assume that P_{i_k} has been placed into the knapsack in a not top-left-packable way (say, as in the placement given by OPT_W). We translate P_{i_k} upward until it intersects the top side of the knapsack at a point (z^*, N). The main idea is to now show that we can *continuously slide* the triangle P_{i_k} by decreasing z while "pivoting" on s, making the packing top-left-packable. Hence, we need to show that the triangle still fits into the knapsack during this continuous sliding process. By Lemma 29, we get that $z^* \in [0, 2\epsilon N]$. Note that P_{i_k} is placed inside the triangle $p(z^*), p^f(z^*), p^g(z^*)$.

Let $f(z) := \|p(z) - p^f(z)\|^2$, $g(z) := \|p(z) - p^g(z)\|^2$, $u(z) = (u_x(z), u_y(z))$ be the placement of the vertex of P_{i_k} that is not adjacent to the longest edge, and $w(z) = (w_x(z), w_y(z))$ be the vertex of P_{i_k} that is not u or $v^*_{i_k}$. Note that it suffices to prove that $f(0) \geq f(z)$ and that $u(z)$ is placed inside the knapsack for each $z \in [0, 2\epsilon N]$ as this implies that P_{i_k} can be placed inside the triangle with vertices $p(0), p^f(0), p^g(0)$, this triangle is contained in $K \setminus H^-$, and this placement is group-respecting.

We begin by proving that $f(0) \geq f(z)$ for every $z \in [0, 2\epsilon N]$. Since $\frac{N}{2} + s^* - z \geq \left(\frac{1}{2} - 2\epsilon\right) N > 0$ and $N - z \geq (1 - 2\epsilon)N > 0$, a similarity argument between triangles $p(z), (z, p^f_y(z)), p^f(z)$ and $p(z), (z, N/2), s$ allows us to obtain

$$\frac{\frac{N}{2}}{\frac{N}{2} + s^* - z} = \frac{N - p^f_y(z)}{N - z}. \tag{5}$$

As $p^f_y(z) \geq 0$, we obtain $N - z \leq N + 2s^* - 2z$, implying that $z \leq 2s^*$. Therefore, we only need to prove that $f(0) \geq f(z)$ for each $z \in [0, \min\{2\epsilon N, 2s^*\}]$. Pythagoras theorem on $p(z), p^f(z), p_{TR}$ gives us

$$f(z) = (N - z)^2 + \left(N - p^f_y(z)\right)^2. \tag{6}$$

Combining (5) and (6), we obtain

$$\begin{aligned} f(z) &= (N - z)^2 + \left(N - p^f_y(z)\right)^2 \\ &= (N - z)^2 \left(1 + \frac{N^2}{(N + 2s^* - 2z)^2}\right). \end{aligned}$$

Note that $f(0) \geq f(2s^*)$ since

$$\begin{aligned} f(0) - f(2s^*) &= N^2 \left(1 + \frac{N^2}{(N + 2s^*)^2}\right) - (N - 2s^*)^2 \left(1 + \frac{N^2}{(N - 2s^*)^2}\right) \\ &= \frac{N^4 - (N + 2s^*)^2(N - 2s^*)^2}{(N + 2s^*)} \\ &= \frac{N^4 - (N^2 - (2s^*)^2)^2}{(N + 2s^*)} \geq 0. \end{aligned}$$

Thus, it is sufficient to show that $-f$ is unimodal in the interval $[0, 2s^*]$—that is, there exists a t such that f is decreasing in $[0, t]$ and increasing in $[t, 2s^*]$. Let $\gamma(z)$ be the angle between $\overline{p(z)p^f(z)}$ and the top of the knapsack, d the length of the line segment $\overline{sp_{TR}}$, and τ be the angle between $\overline{sp_{TR}}$ and the top of the knapsack. By the law of sines, we obtain that

$$-f(z) = -d \left(\frac{\sin(\tau)}{\sin(\gamma(z))} + \frac{\cos(\tau)}{\cos(\gamma(z))}\right).$$

which is unimodal with respect to γ, hence also unimodal in z. Hence, we conclude that $f(z) \leq \max\{f(0), f(2s^*)\} = f(0)$ for $z \in [0, \min\{2\epsilon N, 2s^*\}]$.

Let L be the vertical line that contains $p(z)$, and let $\alpha := \alpha(z)$ be the angle between L and $\overline{p(z)p^f(z)}$. We aim to show that $u(0) \in K$, implying that P_{i_k} is placed inside the knapsack. By examining the triangle $p(z)$, $(z, p^g_y(z))$, $p^g(z)$, we get that

$$\tan(\alpha(z) + \theta) = \frac{\|p^g(z) - (z, p^g_y(z))\|}{\|p(z) - (z, p^g_y(z))\|} = \frac{N - z}{|N - p^g_y(z)|} \geq \frac{(1 - 2\epsilon)N}{|N - p^g_y(z)|}. \tag{7}$$

Furthermore, by the law of sines on triangle $p(z)$, $p^f(z)$, $p^g(z)$, we obtain that

$$\frac{\sin\theta}{\|p^g(z)-p^f(z)\|}=\frac{\sin\alpha(z)}{\|p^g(z)-p(z)\|}.$$

Therefore,

$$\|p^g(z)-p^f(z)\|=\frac{\sin\theta}{\sin(\alpha(z))}|p^g(z)-p(z)|\leq\frac{\epsilon\sqrt{2}}{\frac{\sqrt{2}}{2}-\epsilon}N.$$

By choosing ϵ small enough, we assume that $\|p^g(z)-p^f(z)\|\leq\frac{1}{10}N$ and $\tan(\alpha(z)+\theta)\leq\frac{21}{20}$. Using the second bound on (7), we obtain

$$(1-2\epsilon)N\leq\left(1+\frac{1}{20}\right)|N-p^g(0)|.$$

Again, by choosing $\epsilon>0$ sufficiently small, we guarantee that

$$N\leq\left(1+\frac{1}{10}\right)|N-p^g_y(z)|,$$

which implies that $p^g_y(z)\leq\frac{1}{10}N$ or $p^g_y(z)\geq\frac{21}{10}N$. If $p^g_y(z)\geq\frac{21}{10}N$, then $p^f(z)_y\geq\frac{20}{10}N$, which is not possible. Hence, $p^g_y(z)\leq\frac{1}{10}N<N$, and since $u_y(z)\in\operatorname{conv}(\{p_y(z),p^g_y(z)\})=[p^g_y(z),N]$, we conclude that $0\leq u_y(z)\leq N$. It only remains to prove that the same holds for $u_x(0)$.

Let θ' be the angle of P_{i_k} at w and $\beta(z)$ the angle between $\overline{uw}$ and $\{w_y(z)\}\times[w_x(z),\infty)$. Note that $\beta(z)+\theta'+\pi/2-\alpha(z)=\pi$, and therefore $\alpha'(z)=\beta'(z)$. By examining the triangle $p(z)$, s, $(z,N/2)$, we obtain that

$$\tan(\alpha(z))=\frac{\|s-(z,N/2)\|}{\|p(z)-(z,N/2)\|}=\frac{N+2s^*-2z}{N}=1+2\frac{s^*-z}{N}.$$

Therefore,

$$\beta'(z)=\alpha'(z)=\frac{-\frac{2}{N}}{1+\left(1+2\frac{s^*-z}{N}\right)^2}\leq 0.$$

Since β is decreasing, we conclude that $\beta(0)\geq\beta(z^*)$. Call $\ell'=\|w-u\|$ and $R(\beta)$ the rotation matrix by β, then $u(z)=w(z)+R(\beta(z))(0,\ell')$. In particular,

$$0\leq u_x(0)=w_x(0)-\sin(\beta(0))\ell'\leq w_x(z^*)-\sin(\beta(z^*))\ell'=u_x(z^*)\leq N.\qquad\square$$

Lemma 33. *If each triangle in* OPT_W *is bottom-facing, we have that* $\mathrm{OPT}_W^{(t)}\cup\{P_{i_k}\}$ *is bottom-right-packable.*

Proof. Let $s^*=\sum_{P_{i_\ell}\in\mathrm{OPT}^t_W}\hat{d}_{i_\ell}$, $s^r=(s^*+\frac{N}{2},\frac{N}{2})$ and $s^\ell=(\frac{N}{2}-s^*,\frac{N}{2})$. We define $p(z)$ as (z,N). Define $p^f(z)$ as the intersection between $\overline{p(z)s^r}$ and the bottom edge of K. Similarly, let L^θ be the rotation of $\overline{p(z)s^r}$ by θ counterclockwise, and call $p^g(z)$ the intersection between L^θ and the bottom edge. Let also $T(z)$ be the triangle $p(z)$, $p^f(z)$, $p^g(z)$. We begin by rotating $\mathrm{OPT}_W^{(t)}$ by π around p_M.

We proceed to translate P_{i_k} upward until it intersects the top edge of the knapsack at $p(z^*)$ for some z^*. Note that P_{i_k} is contained inside the triangle $T(z^*)$.

Let I be the amount $T(z^*)$ intersects L. A similarity argument between $T(z^*)$ and (z^*,N), $(z^*,N/2)$, $s^r+(I,0)$ gives us

$$\frac{1}{2}=\frac{N/2+s^*-z^*+I}{p^g(z)-z^*}.$$

Since $p^g(z^*)\leq N$, we obtain $2s+2I\leq z^*$.

We now translate P_{i_k} to the left until $v^*_{i_k}$ coincides with p_{TL}. Let $q = (q_x, q_y)$ be the rightmost point in $P_{i_k} \cap L$. Since $2s + 2I \leq z^*$, we know that $q_x \leq \frac{N}{2} - s$. Therefore, P_{i_k} is placed to the left of the line L^* that passes through v_{TL} and s^ℓ. Furthermore, $\mathrm{OPT}^{(t)}_W$ is to the right of L^* as they are bottom-right-packed. We then rotate P_{i_k} counterclockwise around $v^*_{i_k}$ until it overlaps s^ℓ. Finally, by rotating $\mathrm{OPT}^{(t)}_W \cup \{P_{i_k}\}$ by π around p_M, we arrive at a bottom-right-packing of $\mathrm{OPT}^{(t)}_W \cup \{P_{i_k}\}$. □

We add P_{i_k} to $\mathrm{OPT}^{(t)}_W$. We continue iteratively until we have assigned all triangles in OPT_W to the sets $\mathrm{OPT}^{(1)}_W, \dots, \mathrm{OPT}^{(g)}_W$. Then, the most profitable set $\mathrm{OPT}^{(t^*)}_W$ among them satisfies that $w(\mathrm{OPT}^{(t^*)}_W) \geq \frac{1}{g} w(\mathrm{OPT}_W)$. However, $w(P_{i^*}) \geq \Omega(w(\mathrm{OPT} \cap \mathcal{P}_H \setminus \mathrm{OPT}_W))$. Hence, $w(\mathrm{OPT}^{(t^*)}_W) \geq \frac{1}{g} w(\mathrm{OPT} \cap \mathcal{P}_H)$ or $w(P_{i^*}) \geq \Omega(w(\mathrm{OPT} \cap \mathcal{P}_H))$, which completes the proof of Lemma 23.

2.4.3 Corner-Facing Triangles. In this subsection, we present an $O(1)$-approximation algorithm for the corner-facing triangles in OPT—that is, our algorithm computes a solution $\mathcal{P}' \subseteq \mathcal{P}$ of profit at least $\Omega(w(\mathrm{OPT}_{\mathrm{CF}}))$. We first establish some properties for $\mathrm{OPT}_{\mathrm{CF}}$. We argue that by losing a constant factor, we can assume that each triangle in $\mathrm{OPT}_{\mathrm{CF}}$ intuitively faces the bottom-right corner.

Lemma 34. *By losing a factor 4, we can assume that for each triangle $P_i \in \mathrm{OPT}_{\mathrm{CF}}$ we have that $R^{(1)}_i \setminus \{v^*_i\}$ intersects the bottom edge of the knapsack and $R^{(2)}_i \setminus \{v^*_i\}$ intersects the right edge of the knapsack, or vice versa.*

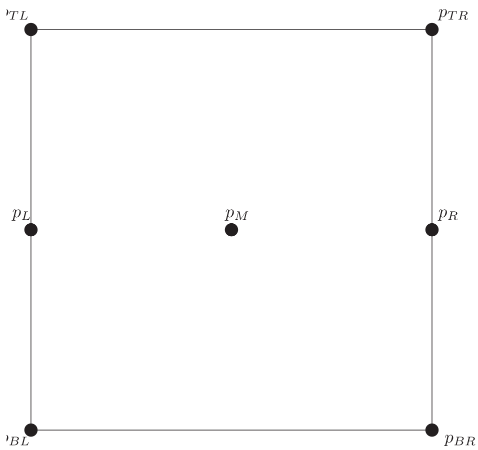
Proof. We can partition $\mathrm{OPT}_{\mathrm{CF}}$ into four groups according to which corner the triangles in this group face. By losing a factor of 4, we keep only the group with largest weight. Then, we rotate the solution appropriately such that the claim of the lemma holds. □

In the following lemma, we establish a property that will be crucial for our algorithm. Let D_i be the ray which starts at p_{BR} and contains v^*_i. For each $P_i \in \mathrm{OPT}_{\mathrm{CF}}$, let R^{ext}_i denote the ray originating at v^*_i which is contained in D_i. We establish that we can assume that R^{ext}_i does not intersect with any triangle $P_{i'} \in \mathrm{OPT}_{\mathrm{CF}}$ (Figure 6).

Lemma 35. *By losing a factor $O(1)$, we can assume that for each $P_i, P_{i'} \in \mathrm{OPT}_{\mathrm{CF}}$ it holds that $R^{\mathrm{ext}}_i \cap P_{i'} = \emptyset$.*

Proof. Let $\epsilon > 0$ be sufficiently small. By losing a factor $O_\epsilon(1)$, we can assume for each triangle win $\mathrm{OPT}_{\mathrm{CF}}$ that its longest edge has length at least $(1-\epsilon)\sqrt{2}N$. This holds since all other triangles are contained in only $O_\epsilon(1)$ groups $\mathcal{P}_j$ with only $O_\epsilon(1)$ triangles in $\mathrm{OPT}_{\mathrm{CF}}$ in total (see Lemma 19). Thus, if these other triangles form a constant fraction of the profit of OPT, then we can obtain an $O(1)$-approximate solution by simply guessing the most profitable triangle among them and a placement for it inside of the knapsack (using that there is a packing in which one vertex of the triangle lies on a corner of the knapsack).

Assume by contradiction that there is a triangle $P_{i'} \in \mathrm{OPT}_{\mathrm{CF}}$ with $R^{\mathrm{ext}}_i \cap P_{i'} \neq \emptyset$. Recall that v^*_i and $v^*_{i'}$ are the vertices of P_i and $P_{i'}$ that are closest to p_{TL}, respectively. Call B the ball centered at p_{TL} with radius $3\epsilon N$. Note that D_i splits B into two parts, one which contains the point $(0, (1-3\epsilon)N)$ and one which does not; let us call these parts B_{left} and B_{right}, respectively. Similarly, D_i splits K into two parts, one which contains p_{BL} and one which does not; let us call these parts K_{left} and K_{right}, respectively. By Lemma 29,, we have that $v^*_i, v^*_{i'} \in B$. Furthermore, assume w.l.o.g. that $v_{i'} \in B_{\mathrm{left}} \subseteq K_{\mathrm{left}}$, as the other case is symmetrical. Let e^1 and e^2 be the two longest edges of $P_{i'}$. Since $P_{i'}$ intersects R^{ext}_i, it is clear that e^1 or e^2 intersect R^{ext}_i.



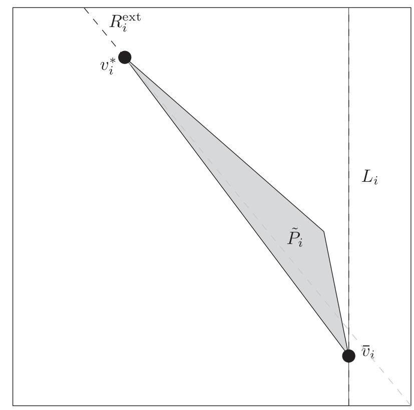


Fig. 6. Left: The points $p_{TL}, p_{TR}, p_{BL}, p_{BR}, p_L, p_M, p_R$. Right: A corner-facing triangle, its vertices v_i^* and $\bar{v}_i$, and the lines L_i and R_i^{ext}.

Assume first that both e^1 and e^2 intersect R_i^{ext}. Let L^1 and L^2 denote the rays originating in $v_{i'}^*$ and containing e^1 and e^2, respectively. Then both of them do not intersect the bottom edge of K since otherwise they would intersect D_i in two different points, which is impossible since $L^1 \neq D_i \neq L^2$ and L^1, L^2, and D_i are rays. Therefore, $P_{i'}$ is not corner-facing, which is a contradiction.

Assume now that exactly one edge among e^1 and e^2 intersects R_i^{ext}. Since the lengths of both e^1 and e^2 are greater than $(\sqrt{2}-\epsilon)N/2$ and $v_i^* \in B'$, if ϵ is sufficiently small enough we conclude that $v_i^* \in P_{i'}$, which is a contradiction. □

Our algorithm is a DP that intuitively guesses the placements of the triangles in OPT_{CF} step by step. To this end, each DP-cell corresponds to a subproblem that is defined via a part $K' \subseteq K$ of the knapsack and a subset of the groups $J \subseteq \{j_{\min}, \ldots, j_{\max}\}$. The goal is to place triangles from $\bigcup_{j \in J} \mathcal{P}_j$ of maximum profit into K'. Formally, each DP-cell is defined by up to two triangles $P_i, P_{i'}$, placements $\tilde{P}_i, \tilde{P}_{i'}$ for them, and a set $J \subseteq \{j_{\min}, \ldots, j_{\max}\}$; if the cell is defined via exactly one triangle P_i, then there is also a value dir $\in$ {left, mid}. The corresponding region K' is defined as follows: if the cell is defined via zero triangles, then the region is the whole knapsack K. Otherwise, let $\bar{v}_i$ denote the right-most vertex of $\tilde{P}_i$—that is, the vertex of $\tilde{P}_i$ that is closest to the right edge of the knapsack (see Figure 6). Let L_i denote the vertical line that goes through $\bar{v}_i$ (and thus intersects the top and the bottom edge of the knapsack). If the cell is defined via one triangle P_i, then observe that $K \setminus (\tilde{P}_i \cup R_i^{\text{ext}} \cup L_i)$ has three connected components:

— one on the left, surrounded by R_i^{ext}, parts of $\tilde{P}_i$, the left edge of the knapsack, and parts of the top and bottom edge of the knapsack,
— one on the right, surrounded by L_i, the right edge of the knapsack, and parts of the top and bottom edge of the knapsack, and
— one in the middle, surrounded by the top edge of the knapsack, $\tilde{P}_i$, R_i^{ext}, and L_i.

If dir = left, then the region of the cell equals the left component, and if dir = mid, then the region of the cell equals the middle component. Assume now that the cell is defined via two triangles $P_i, P_{i'}$. Furthermore, assume w.l.o.g. that $\bar{v}_i$ is closer to the right edge of the knapsack than $\bar{v}_{i'}$. Then, $K \setminus (\tilde{P}_i \cup \tilde{P}_{i'} \cup R_i^{\text{ext}} \cup R_{i'}^{\text{ext}} \cup L_{i'})$ has one connected component that is surrounded by

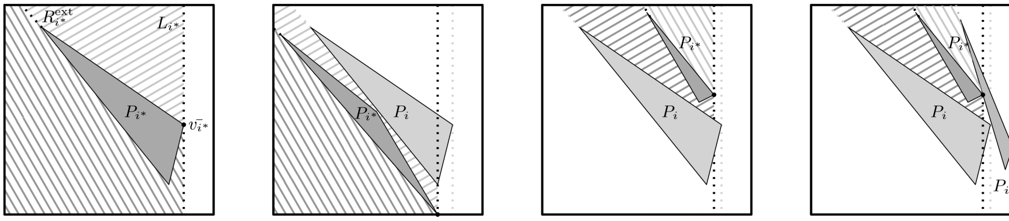


Fig. 7. The cases in the transition of the DP for corner-facing triangles (see Lemma 36).

$\tilde{P}_i, \tilde{P}_{i'}, R_i^{\text{ext}}, R_{i'}^{\text{ext}}, L_{i'}$ and we define the region of the cell to be this component. Observe that the total number of DP-cells is bounded by $(nN)^{O(1)}$, using that there are only $(nN)^{O(1)}$ possible placements for each triangle.

We describe a DP that computes the optimal solution to each cell. Assume that we are given a cell C for which we want to compute the optimal solution. We guess the triangle P_{i^*} in the optimal solution to this cell such that $\bar{v}_{i^*}$ is closest to the right edge of the knapsack, and its placement $\tilde{P}_{i^*}$ in the optimal solution to C. Let j^* such that $P_{i^*} \in \mathcal{P}_{j^*}$. We will prove in the next lemma that the optimal solution to C consists of P_{i^*} and the optimal solutions to two other DP-cells (Figure 7).

Lemma 36. *Let C be a DP-cell, let $J \subseteq \{j_{\min}, \dots, j_{\max}\}$, and let $P_i \in \mathcal{P}_\ell, P_{i'} \in \mathcal{P}_{\ell'}$ be two triangles with $\ell < \ell'$ and let $\tilde{P}_i, \tilde{P}_{i'}$ be placements for them. Then, there are disjoint sets $J', J'' \subseteq J$ such that*

1. *if $C = (J)$, then its optimal solution consists of P_{i^*} and the optimal solutions to the cells $(J', P_{i^*}, \tilde{P}_{i^*}, \text{left})$ and $(J'', P_{i^*}, \tilde{P}_{i^*}, \text{mid})$,*
2. *if $C = (J, P_i, \tilde{P}_i, \text{left})$, then its optimal solution consists of P_{i^*} and the optimal solutions to the cells $(J', P_{i^*}, \tilde{P}_{i^*}, \text{left})$ and $(J'', P_i, \tilde{P}_i, P_{i^*}, \tilde{P}_{i^*})$,*
3. *if $C = (J, P_i, \tilde{P}_i, \text{mid})$, then its optimal solution consists of P_{i^*} and the optimal solutions to the cells $(J', P_{i^*}, \tilde{P}_{i^*}, \text{mid})$ and $(J'', P_i, \tilde{P}_i, P_{i^*}, \tilde{P}_{i^*})$,*
4. *if $C = (J, P_i, \tilde{P}_i, P_{i'}, \tilde{P}_{i'})$, then the optimal solution to C consists of P_{i^*} and the optimal solutions to the cells $(J', P_i, \tilde{P}_i, P_{i^*}, \tilde{P}_{i^*})$ and $(J'', P_{i^*}, \tilde{P}_{i^*}, P_{i'}, \tilde{P}_{i'})$.*

Proof. Let OPT_C denote the optimal solution to the cell C. The following claim is easy to prove.

Claim 2. *Consider a feasible solution S for the cell C. Let P_{i^*} be the triangle in S whose vertex $\bar{v}_{i^*}$ is closest to the right edge of the knapsack. Then, it holds that $L_{i^*} \cap P_i = \emptyset$ for each triangle $P_i \in S$.*

First assume that $C = (J)$. By Lemmas 35 and Claim 2, no triangle in OPT_C intersects L_{i^*} or $R_{i^*}^{\text{ext}}$ and no triangle in OPT_C has a vertex on the right of L_{i^*}. Hence, each triangle in OPT_C is contained in the area corresponding to the cells $(J', P_{i^*}, \tilde{P}_{i^*}, \text{left})$ and $(J'', P_{i^*}, \tilde{P}_{i^*}, \text{mid})$. We define J' to be the set of indices $j \in J$ such that in OPT_C there is a triangle $P_i \in \text{OPT}_C$ contained in the area corresponding to $(J', P_{i^*}, \tilde{P}_{i^*}, \text{left})$ and J'' similarly. The other cases can be verified similarly. □

We guess the sets $J', J'' \subseteq J$ according to Lemma 36 and store in C the solution consisting of P_{i^*}, and the solutions stored in the two cells according to the lemma. At the end, the cell $C = (\{j_{\min}, \dots, j_{\max}\})$ (whose corresponding region equals K) contains the optimal solution.

Lemma 37. *There is an algorithm with a running time of $(nN)^{O(1)}$ that computes a solution $\mathcal{P}' \subseteq \mathcal{P}$ such that $w(\text{OPT}_{\text{CF}})=O(w(\mathcal{P}'))$.*

Proof. Since we applied Lemma 20, there are only $(nN)^{O(1)}$ different placements for each triangle. Additionally, there are only $2^{O(\log N)} = N^{O(1)}$ possibilities for the set J in the description of the DP-cell. Therefore, the number of DP-cells is bounded by $(nN)^{O(1)}$. To compute the value of a DP-cell C, we guess the triangle P_{i^*} and its corresponding placement $\tilde{P}_{i^*}$, and in particular, we reject a guess if $\tilde{P}_{i^*}$ is not contained in the region corresponding to C. Additionally, we guess J' and J'' for which there are only $N^{O(1)}$ possibilities each and reject guesses which do not satisfy that $J' \subseteq J$, $J'' \subseteq J$, and that $J' \cap J'' = \emptyset$. Therefore, in each DP-cell, we store a solution that is feasible. We can fill the complete DP-table in time $(nN)^{O(1)}$. Using Lemma 36, one can show that the cell $C = (\{j_{\min}, \ldots, j_{\max}\})$ contains a solution $\mathcal{P}'$ with weight at least $\Omega(w(\mathrm{OPT}_{\mathrm{CF}}))$. □

By combining Lemmas 25 and 37, we obtain the proof of Lemma 9.

2.5 Hard Polygons under Resource Augmentation

Let $\delta > 0$. We consider the setting of $(1 + \delta)$-resource augmentation—that is, we want to compute a solution $\mathcal{P}' \subseteq \mathcal{P}$ that is feasible for a knapsack of size $(1 + \delta)N \times (1 + \delta)N$ and such that $w(\mathrm{OPT}){=}O(w(\mathcal{P}'))$ where OPT is the optimal solution for the original knapsack of size $N \times N$. Note that increasing K by a factor of $1 + \delta$ is equivalent to shrinking the input polygons by a factor of $1 + \delta$.

Given a polygon P defined via coordinates $(x_1, y_1), \ldots, (x_k, y_k) \in \mathbb{R}^2$, we define $\mathrm{shr}_{1+\delta}(P)$ to be the polygon with coordinates $(\bar{x}_1, \bar{y}_1), \ldots, (\bar{x}_k, \bar{y}_k) \in \mathbb{R}^2$ where $\bar{x}_{k'} = x_{k'}/(1+\delta)$ and $\bar{y}_{k'} = y_{k'}/(1+\delta)$ for each k'. For each input polygon $P_i \in \mathcal{P}$, we define its shrunk counterpart to be $\bar{P}_i := \mathrm{shr}_{1+\delta}(P_i)$. Let $\bar{\ell}_i$ denotes the length of the diametrical segment of $\bar{P}_i$. Based on the corresponding value of $\bar{\ell}_i$, we define sets $\bar{\mathcal{P}}_E, \bar{\mathcal{P}}_M, \bar{\mathcal{P}}_H$ and the set $\bar{\mathcal{P}}_j$ for each $j \in \mathbb{Z}$ in the same way as we defined $\mathcal{P}_E, \mathcal{P}_M, \mathcal{P}_H$ and $\mathcal{P}_j$ based on ℓ_i above. Therefore, these definitions are based of each polygon P_i *after* shrinking P_i by a factor of $1 + \delta$.

For the sets $\bar{\mathcal{P}}_E$ and $\bar{\mathcal{P}}_M$, we use the algorithms due to Lemmas 6 and 7 as before. Recall that we may assume that in the original polygons there are no polygons with diameter greater than $\sqrt{2}N$. For the hard polygons $\bar{\mathcal{P}}_H$, we can show that there are only $O_\delta(1)$ groups $\bar{\mathcal{P}}_j$ that are non-empty, using that we obtained them via shrinking the original input polygons. Intuitively, this is true since $\bar{\ell}_i \leq \frac{\sqrt{2}N}{1+\delta}$ for each $\bar{P}_i \in \bar{\mathcal{P}}$ and hence $\bar{\mathcal{P}}_j \cap \bar{\mathcal{P}}_H = \emptyset$ if $j < \log(\frac{\delta}{1+\delta}\sqrt{2}N)$.

Lemma 38. *We have that $\bar{\mathcal{P}}_j = \emptyset$ if $j < \log(\frac{\delta}{1+\delta}\sqrt{2}N)$. Hence, there are only $\log(\frac{1+\delta}{\delta}) + 1$ values $j \in \mathbb{Z}$ such that $\bar{\mathcal{P}}_j \neq \emptyset$.*

Proof. Let P_i be an arbitrary polygon. Note that $\bar{\ell}_i = \frac{1}{1+\delta}\ell_i$ and therefore $\bar{\ell}_i \leq \frac{\sqrt{2}N}{1+\delta}$. We conclude that $\frac{\delta}{1+\delta}\sqrt{2}N \leq \sqrt{2}N - \bar{\ell}_i \leq \sqrt{2}N$. Note that for any j, if $\bar{\mathcal{P}}_j$ is non-empty, there must be a P_i that satisfies $\sqrt{2}N - 2^j \leq \bar{\ell}_i < \sqrt{2}N - 2^{j-1}$ (or equivalently, $2^{j-1} < \sqrt{2}N - \bar{\ell}_i \leq 2^j$). We conclude that such j's must satisfy $\log(\frac{\delta}{1+\delta}\sqrt{2}N) \leq j < \log(2\sqrt{2}N)$, and therefore there are at most $\log(2\sqrt{2}N) - \log(\frac{\delta}{1+\delta}\sqrt{2}N) = \log(\frac{1+\delta}{\delta}) + 1$ non-empty $\bar{\mathcal{P}}_j$. □

Lemmas 10, 19, and 38 imply that $|\overline{\mathrm{OPT}} \cap \bar{\mathcal{P}}_H| {=} O(\log(\frac{1+\delta}{\delta}))$, where $\overline{\mathrm{OPT}}$ denotes the optimal solution for the polygons in $\bar{\mathcal{P}}$. Let $\bar{\mathcal{P}}'_H \subseteq \bar{\mathcal{P}}_H$ denote the set due to Lemma 20 when assuming that $\bar{\mathcal{P}}_H$ are the hard polygons in the given instance. Therefore, we guess $\bar{\mathcal{P}}'_H$ in time $n^{O(\log(\frac{1+\delta}{\delta}))}$. Finally, we output the solution of largest weight among $\bar{\mathcal{P}}'_H$ and the solutions due to applying Lemmas 6 and 7 to the input sets $\bar{\mathcal{P}}_E$ and $\bar{\mathcal{P}}_M$, respectively. This yields the proof of Theorem 3.

3 OPTIMAL PROFIT UNDER RESOURCE AUGMENTATION

In this section, we also study the setting of $(1 + \delta)$-resource augmentation—that is, we want to compute a solution $\mathcal{P}'$ which is feasible for an enlarged knapsack of size $(1+\delta)N \times (1+\delta)N$, for any constant $\delta > 0$. We present an algorithm with a running time of $n^{(\log(n)/\delta)^{O(1)}}$ that computes such a solution $\mathcal{P}'$ with $w(\mathcal{P}') \geq w(\text{OPT})$, where OPT is the optimal solution for the original knapsack of size $N \times N$. In particular, we here do not lose any factor in our approximation guarantee.

In Section 3.1, we will prove a set of properties that we can assume "by $(1+\delta)$-resource augmentation," meaning that if we increase the size of K by a factor $1+\delta$, then there exists a solution of weight $w(\text{OPT})$ with the mentioned properties, or that we can modify the input in time $n^{O(1)}$ such that it has these properties and there still exists a solution of weight $w(\text{OPT})$. We will perform several such operations and hence at the end increase the knapsack by a factor $(1+\delta)^{O(1)} = (1+O(\delta))$. We then obtain our claimed result by scaling δ appropriately.

In Section 3.2, we will present our separator-based recursive algorithm. Roughly speaking, we guess a balanced separator and then partition the problem into two subproblems: one inside the separator and one outside the separator.

3.1 Few Types of Items

We want to establish that the input polygons have only $(\log(n)/\delta)^{O(1)}$ different shapes. Like in Section 2, for each polygon $P_i \in \mathcal{P}$, denote by B_i its bounding box with width ℓ_i and height h_i. Note that $\ell_i \geq h_i$.

First, we argue that if we increase the size of the knapsack, then there exists a packing in which any two polygons have a distance of at least $\Omega(\delta N/n)$. This will allows us later to modify our polygons in this packing.

Lemma 39. *By $(1+O(\delta))$-resource augmentation, we can assume that the distance between any two polygons $P_i, P_{i'} \in \text{OPT}$ is at least $\Omega(\delta N/n)$.*

Proof. Assume that $\text{OPT} = \{P_1, \ldots, P_k\}$. For each polygon $P_i \in \text{OPT}$, denote by $\tilde{P}_i$ its corresponding placement in OPT. We assume that for any $\tilde{P}_i, \tilde{P}_{i'}$ with $i < i'$ it holds that intuitively $\tilde{P}_i$ lies on the left of $\tilde{P}_{i'}$. Formally, we require that if there is a horizontal line L that has non-empty intersection with both $\tilde{P}_i$ and $\tilde{P}_{i'}$, then $L \cap \tilde{P}_i$ lies on the left of $L \cap \tilde{P}_{i'}$.

We need to prove that such an ordering indeed exists. We consider in the proof of this the closure of our polygons, which makes our requirement for the horizontal lines L only stronger. We define a relation $\prec$ such that $\tilde{P}_i \prec \tilde{P}_{i'}$ if there is a horizontal line L that has non-empty intersection with both $\tilde{P}_i$ and $\tilde{P}_{i'}$ such that $L \cap \tilde{P}_i$ lies on the left of $L \cap \tilde{P}_{i'}$. We want to show that $\prec$ is a partial order. Suppose that it is not a partial order. Consider an instance with the smallest number of polygons in which $\prec$ is not a partial order. Then there are polygons $\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_k, \tilde{P}_{k+1}$ with $\tilde{P}_{k+1} = \tilde{P}_1$ such that $\tilde{P}_i \prec \tilde{P}_{i+1}$ for each $i \in \{1, \ldots, k\}$. For each $i \in \{1, \ldots, k\}$, let L_i be a horizontal line having non-empty intersection with $\tilde{P}_i$ and $\tilde{P}_{i+1}$ such that $L \cap \tilde{P}_i$ lies on the left of $L \cap \tilde{P}_{i+1}$. Since our example has a minimum number of polygons, for each $i \in \{1, \ldots, k\}$ there is no line L intersecting $\tilde{P}_i$ and some polygon $\tilde{P}_{i'}$ with $i' > i+1$ such that $L \cap \tilde{P}_i$ lies on the left of $L \cap \tilde{P}_{i'}$. Assume w.l.o.g. that L_1 has a larger y-coordinate than L_2—that is, L_1 is higher than L_2. We move each line L_i downward as much as possible. By the minimality of our instance, L_1 still has a larger y-coordinate than L_2; otherwise, there is a line intersecting $\tilde{P}_1$, $\tilde{P}_2$, and $\tilde{P}_3$. Furthermore, for each $i \in \{1, \ldots, k\}$, the line L_i has a larger y-coordinate than the line L_{i+1}. Therefore, the line L_k does not intersect $\tilde{P}_1$, which is a contradiction.

Now for each $k' \in \{1, \ldots, k\}$, we move $\tilde{P}_{k'}$ by $k' \cdot \frac{\delta}{n} N$ units to the right. Since $k \leq n$, the resulting placement fits into the knapsack using $(1+\delta)$-resource augmentation. Intuitively, in the resulting placement, each polygon $\tilde{P}_i$ has $\frac{\delta}{n} N$ units of empty space on its left and on its right. In a similar fashion, we move all polygons up such that they still fit into the knapsack under $(1+\delta)$-resource augmentation and, intuitively, each polygon has $\frac{\delta}{n} N$ units of empty space above and

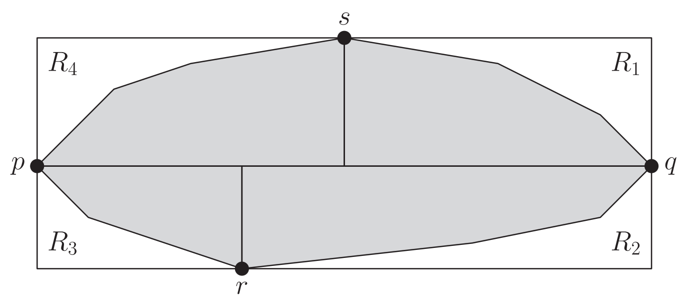


Fig. 8. The points and rectangles used in the proof of Lemma 41.

below it. Therefore, in the resulting packing, the distance between any two polygons is at least $\Omega(\delta N/n)$. □

We invoke Lemma 39 and use the gained space around each polygon $P_i \in \mathcal{P}$ to enlarge it such that $\ell_i \geq h_i \geq \delta N/n$ and $\text{area}(P_i) \geq \Omega(\delta^2/n^2 \text{area}(K))$

Lemma 40. *By $(1 + O(\delta))$-resource augmentation, we can assume for each $P_i \in \mathcal{P}$ that $\ell_i \geq h_i \geq \delta N/n$ and that* $\text{area}(P_i) \geq \frac{\delta^2}{2n^2}\text{area}(K)$.

Proof. If for a polygon $P_i \in \mathcal{P}$ it holds that $h_i < \delta N/n$, then we replace P_i by a rectangle of height $\delta N/n$ and width $\max\{\ell_i, \delta N/n\}$. After this modification, it holds that $\text{area}(P_i) \geq \frac{1}{2}\ell_i h_i = \frac{1}{2}\frac{\delta^2 N^2}{n^2}$ for each remaining polygon P_i. By Lemma 39, the enlarged polygons from OPT still fit into the knapsack using $(1 + O(\delta))$-resource augmentation. □

Next, intuitively we stretch the optimal solution OPT by a factor $1 + \delta$ which yields a container C_i for each polygon $P_i \in \text{OPT}$ which contains P_i and which is slightly bigger than P_i. We define a polygon P_i' such that $P_i \subseteq P_i' \subseteq C_i$ and that globally there are only $(\log(n)/\delta)^{O_\delta(1)}$ different ways P_i' can look like, up to translations and rotations. We refer to those as a set $\mathcal{S}$ of *shapes* of input objects. Hence, due to the resource augmentation, we can replace each input polygon P_i by one of the shapes in $\mathcal{S}$.

Lemma 41. *By $(1 + \delta)$-resource augmentation, we can assume that there is a set of shapes $\mathcal{S}$ with $|\mathcal{S}| \leq (\log(n)/\delta)^{O_\delta(1)}$ such that for each $P_i \in \mathcal{P}$ there is a shape $S \in \mathcal{S}$ such that $P_i = S$ and S has only $\Lambda = (1/\delta)^{O(1)}$ many vertices.*

Proof. Let P be an input polygon. Assume that P is rotated such that the diametrical segment is horizontal—that is, let $p = (p_x, p_y), q = (q_x, q_y)$ denote the vertices of P with largest distance; we assume that p and q lie on a horizontal line (i.e., that $p_y = q_y$). Furthermore, let $r = (r_x, r_y), s = (s_x, s_y)$ denote the vertices of P with minimum and maximum y-coordinate, respectively (Figure 8).

Note that extending the knapsack by a factor $1+\delta$ is equivalent to shrinking each input polygon by a factor $1+\delta$. Let $\text{shr}_{1+\delta}(P)$ denote the polygon obtained by shrinking P toward the origin—that is, by replacing each vertex $v = (v_x, v_y)$ of P by the vertex $v' = (\frac{v_x}{1+\delta}, \frac{v_y}{1+\delta})$. Our goal is to show that there exists a polygon P' whose shape is one shape out of $(\log(n)/\delta)^{O_\delta(1)}$ options such that there is a translation vector $\vec{a}$ with $\vec{a} + \text{shr}_{1+\delta}(P) \subseteq P' \subseteq P$. Then, we replace in the input the polygon P by a polygon $\tilde{P}$ which is congruent to P'. Hence, for the shapes of the resulting polygons $\tilde{P}$, there are only $(\log(n)/\delta)^{O_\delta(1)}$ options. In the process, we will shrink P a constant number of times. Then, the claim follows by redefining δ accordingly.

First, we shrink P by a factor of at most $1+\delta$ such that the line segment connecting p and q has a length that is a power of $1+\delta$. Let ℓ_i' denote this new length. Since originally $\sqrt{2}N \geq \ell_i \geq \delta N/n$, there are only $O(\log_{1+\delta} n)$ options for ℓ_i'. We partition the bounding box of P into four rectangles where

— R_1 is the (unique) rectangle with vertices s and q,
— R_2 is the (unique) rectangle with vertices r and q,
— R_3 is the (unique) rectangle with vertices p and r, and
— R_4 is the (unique) rectangle with vertices p and s.

We translate P such that p is the origin. If the width of R_1 is smaller than $\delta\ell_i'$, then intuitively we shrink P by a factor $1+\delta$ toward p such that q_x is again a power of $1+\delta$ and $s_x = q_x$. First, we move q toward p such that q_x is the next smaller power of $1+\delta$. Then we move s toward p such that $s_x = q_x$. Finally, we move each remaining vertex v by exactly a factor $1+\delta$ toward p. As a result, R_1 becomes empty. We perform similar operations in case that the width of R_2, R_3, or R_4 is smaller than $\delta\ell_i'$. Additionally, we perform a similar operation in case that the height of R_1 (identical to the height of R_4) is smaller than δh_i or that the height of R_2 (identical to the height of R_3) is smaller than δh_i. In the latter operations, we move the vertices of P toward s or r, respectively.

Assume again that p is the origin. Let $t := (s_x, p_y)$. Let $t' = (t_x', t_y')$ such that $t_y' = p_y$, and t_x' is the smallest value t_x' with $t_x' \geq t_x$ such that the distance between t' and q is a multiple of $\delta^3\ell_i'$. In particular, then $t_x' - t_x \leq \delta^3\ell_i' \leq \delta^2(q_x - t_x)$ and note that $q_x - t_x$ is the width of R_1 for which $q_x - t_x \geq \delta\ell_i'$ holds.

We define $s' = (s_x', s_y')$ such that $s_x' = t_x'$ and s_y' is that largest value s_y' such that $s' = (s_x', s_y')$ lies inside P. Observe that $s_y' \geq (1-\delta^2)s_y$ since P includes all points on the line segment connecting s and q by convexity. Similarly, we define a point t'' between p and t and a corresponding point s''. We move each vertex $v = (v_x, v_y)$ of P toward t that satisfy that $t_x'' \leq v_x \leq t_x'$ and $v_y \geq t_y'' = t_y = t_y'$—that is, we reduce the distance between v and t by a factor $1/(1+\delta)$ which we justify via shrinking. One can show that afterward v lies in the convex hull spanned by the other vertices of P and s' and s'', using that $s_y' \geq (1-\delta^2)s_y \geq (1-\delta^2)v_y$. Hence, we can remove v.

We move P such that t' becomes the origin. Let R_1' denote the (unique) rectangle with vertices s' and q. Our goal is now to move the vertices within R_1' such that only $O_\delta(1)$ vertices remain and that for the coordinate of each of them there are only $(\log(n)/\delta)^{O_\delta(1)}$ options. Whenever we move a vertex v within R_1', we move v toward t' such that the distance between v and t' decreases by at most a factor $1+\delta$ but keep s' and q unchanged. Let h denote the distance between t' and s', and let w denote the distance between t' and q. Assume w.l.o.g. that $h = w = 1$ and that t' is the origin. Observe that by convexity each point on the line segment connecting s' and q lies within P.

Let $k \in \mathbb{N}$ be a constant with $k = O_\delta(1)$ to be defined later. We shoot rays $r_0, \ldots, r_k$ originating at t' such that r_0 goes through s', r_k goes through q, and between any two consecutive rays r_j, r_{j+1} there is an angle of exactly $\frac{\pi}{4k}$ (Figure 9). For each $j \in \{0, \ldots, k\}$, denote by v_j the point on the boundary of P that is intersected by r_j (not necessarily a vertex of P). Imagine that we shrink P such that we move each vertex $v = (v_x, v_y)$ of P toward $t' = 0$—that is, we replace v by the point $v' := (\frac{v_x}{1+\delta}, \frac{v_y}{1+\delta})$. We argue that v' lies in the convex hull of $t', v_0, \ldots, v_k$ and hence we can remove v'. Let r' be a ray originating at t' and going through v. Suppose that r_j and r_{j+1} are the rays closest to r'. Let $v_j = (v_j^x, v_j^y)$ and $v_{j+1} = (v_{j+1}^x, v_{j+1}^y)$. Then v lies in the convex hull of v_j, v_{j+1} and the point (v_{j+1}^x, v_j^y). We have that $v_j^x \geq 1/3$ and $v_{j+1}^x \geq 1/3$ or that $v_j^y \geq 1/3$ and $v_{j+1}^y \geq 1/3$. Assume w.l.o.g. that $v_j^x \geq 1/3$ and $v_{j+1}^x \geq 1/3$. We claim that then $v_j^x \leq v_x \leq v_{j+1}^x \leq (1+\delta)v_j^x$. The first two inequalities follow from convexity. For proving that $v_{j+1}^x \leq (1+\delta)v_j^x$, we can assume that $v_j^x \leq 1/(1+\delta)$ since otherwise the claim is immediate. This implies that $v_j^y \geq \Omega(\delta)$. Also

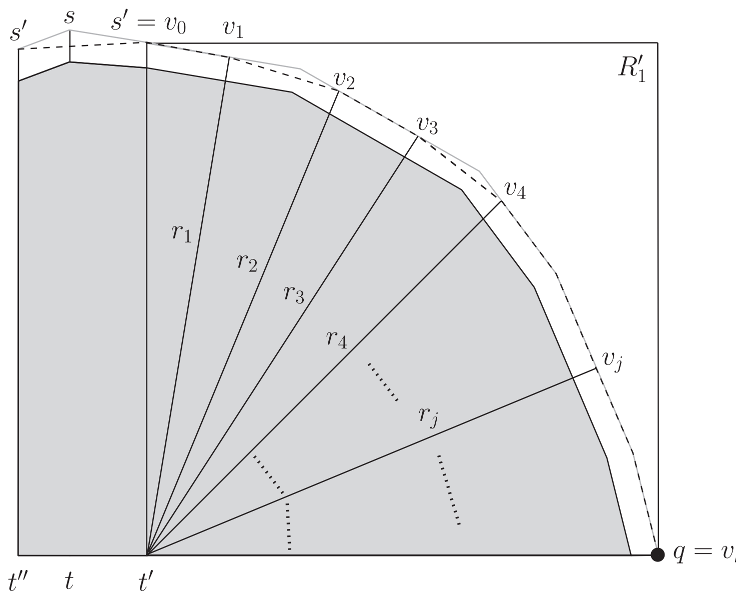


Fig. 9. The points and rays related to the shrinking of R_1. In gray, we see the shrunk polygon $\bar{P}_i$; in gray outline, we see the original polygon P_i; and inside the dashed line, we see the corresponding shape in $\mathcal{S}$.

observe that $v^y_{j+1} \le v^y_j$, since otherwise v_j would not be on the boundary of P, by convexity. Therefore, v^x_{j+1} cannot be larger than the x-coordinate of the point on r_{j+1} with y-coordinate v^y_j. Using that $1/(1+\delta) \ge v^x_j \ge 1/3$, one can show that there is a choice for $k \in O_\delta(1)$ that ensures that $v^x_{j+1} \le (1+\delta)v^x_j$. Finally, for each point v_j with $j \in \{1,\ldots,k-1\}$, we move v_j toward t' such that the distance between v_j and t' becomes an integer power of $1+\delta$. Since before the shrinking this distance was $\Omega(1)$, there are only $O_\delta(1)$ options for the resulting distance.

In a similar way, we define R'_2, R'_3, and R'_4 and perform a symmetric operation on them. The resulting polygon is defined via ℓ'_i, the positions of t', t'', the positions of the vertices u' and u'' (which are defined analogously to t' and t''), for R'_1 the distance between t' and s' and the distances of the $O_\delta(1)$ vertices v_j to t', and the respective values for R'_2, R'_3, and R'_4. For each of these values, there are only $(\log(n)/\delta)^{O_\delta(1)}$ options and there are $O_\delta(1)$ such values in total. Hence, there are $(\log(n)/\delta)^{O_\delta(1)}$ possibilities for the resulting shape. □

Finally, we ensure that for each polygon $P_i \in \mathcal{P}$, we can restrict ourselves to only $(n/\delta)^{O(1)}$ possible placements in K.

Lemma 42. *By $(1+\delta)$-resource augmentation, for each polygon $P_i \in \mathcal{P}$ we can compute a set $\mathcal{L}_i$ of at most $(n/\delta)^{O(1)}$ possible placements for P_i in time $(n/\delta)^{O(1)}$ such that if $P_i \in$ OPT, then in OPT the polygon P_i is placed inside K according to one placement $\tilde{P}_i \in \mathcal{L}_i$.*

Proof. First, we prove that for each polygon P_i, it suffices to allow only $(n/\delta)^{O(1)}$ possible vectors d when defining its placement $\tilde{P}_i$ as $\tilde{P}_i = d + \mathrm{rot}_\alpha(P_i)$. We invoke Lemma 39 such that any two polygons have distance of at least $\frac{\delta}{n}N$. For each polygon P_i, let v_i be its first vertex $(x'_{i,1}, y'_{i,1})$. We move each polygon $\tilde{P}_i$ such that v_i is placed on a point whose coordinates are integral multiples of $\frac{\delta}{4n}N$. For achieving this, it suffices to move $\tilde{P}_i$ by at most $\frac{\delta}{4n}N$ units down and by at most $\frac{\delta}{4n}N$ units to the left. This implies that each polygon is translated by at most $\frac{\sqrt{2}\delta}{4n}N$ units.

Consequently, since the polygons were $\frac{\delta}{n}N$ units apart, after translation every pair of polygons is at least $\frac{\delta}{n}N - 2\frac{\sqrt{2}\delta}{4n}N = (4-2\sqrt{2})\frac{\delta}{4n}N > \frac{\delta}{4n}N$ units apart; in particular, they do not intersect.

We want to argue that we can rotate each polygon $\tilde{P}_i$ such that its angle is one out of $(n/\delta)^{O(1)}$ many possible angles. Consider a polygon $\tilde{P}_i$. Suppose that we rotate it around its vertex v_i. We want to argue that if we rotate $\tilde{P}_i$ by an angle of at most $\frac{\delta}{16n}$, then this moves each vertex of $\tilde{P}_i$ by at most $\frac{\delta}{4n}N$ units. Let v_i' be a vertex of $\tilde{P}_i$ with $v_i \neq v_i'$. Let x denote the distance between the old and the new position of v_i' if we rotate $\tilde{P}_i$ by an angle of α. Then we have that $x = \sin\alpha \frac{\overline{v_i v_i'}}{\sin((\pi-\alpha)/2)} \leq 4N\alpha \leq \frac{\delta}{4n}N$, assuming that $\alpha \leq \frac{\delta}{16n}$ and that δ is sufficiently small.

Therefore, we rotate $\tilde{P}_i$ around v_i by an angle of at most $\frac{\delta}{16n}$ such that $d + \text{rot}_\alpha(P_i) = \tilde{P}_i$ for an angle α which is an integral multiple of $\frac{\delta}{16n}$. Due to our movement of $\tilde{P}_i$ before, we can assume that $d = (d_1, d_2)$ satisfies that d_1 and d_2 are integral multiples of $\frac{\delta}{4n}N$. Thus, for d and for α, there are only $(n/\delta)^{O(1)}$ possibilities, which yields only $(n/\delta)^{O(1)}$ possible placements for P_i. □

3.2 Recursive Algorithm

We describe our main algorithm. First, we guess how many polygons of each of the shapes in $\mathcal{S}$ are contained in OPT. Since there are only $(\log(n)/\delta)^{O_\delta(1)}$ different shapes in $\mathcal{S}$, we can do this in time $n^{(\log(n)/\delta)^{O_\delta(1)}}$. Once we know how many polygons of each shape we need to select, it is clear which polygons we should take, since if for some shape $S_i \in \mathcal{S}$ we need to select n_i polygons with that shape, then it is optimal to select the n_i polygons in $\mathcal{P}$ of shape S_i with largest weight. Therefore, in the sequel, assume that we are given a set of polygons $\mathcal{P}' \subseteq \mathcal{P}$ and we want to find a packing for them inside K.

Our algorithm is recursive, and it generalizes a similar algorithm for the special case of axis-parallel rectangles in [1]. On a high level, we guess a partition of K given by a separator Γ which is a polygon inside K. It has the property that at most $\frac{2}{3}|\text{OPT}|$ of the polygons of OPT lie inside Γ and at most $\frac{2}{3}|\text{OPT}|$ of the polygons of OPT lie outside Γ. We guess how many polygons of each shape are placed inside and outside Γ in OPT. Then, we recurse separately inside and outside Γ. For our partition, we are looking for a polygon Γ according to the following definition.

Definition. Let $\ell \in \mathbb{N}$ and $\epsilon > 0$. Let $\bar{\mathcal{P}}$ be a set of pairwise non-overlapping polygons in K. A polygon Γ is a *balanced $\hat{\epsilon}$-cheap ℓ-cut* if

- Γ has at most ℓ edges,
- the polygons contained in Γ have a total area of at most $2/3 \cdot \text{area}(\bar{\mathcal{P}})$,
- the polygons contained in the complement of Γ (i.e., in $K \setminus \Gamma$) have a total area of at most $2/3 \cdot \text{area}(\bar{\mathcal{P}})$, and
- the polygons intersecting the boundary of Γ have a total area of at most $\hat{\epsilon} \cdot \text{area}(\bar{\mathcal{P}})$.

To restrict the set of balanced cheap cuts to consider, we will allow only polygons Γ such that each of its vertices is contained in a set Q of size $(n/\delta)^{O(1)}$ defined as follows. We fix a triangulation for each placement $P_i' \in \mathcal{L}_i$ of each polygon $P_i \in \mathcal{P}'$. We define a set Q_0 where for each placement $P_i' \in \mathcal{L}_i$ for P_i we add to Q_0 the positions of the vertices of P_i'. Additionally, we add the four corners of K to Q_0. Let $\mathcal{V}$ denote the set of vertical lines $\{(\bar{x}, \bar{y}) | \bar{y} \in \mathbb{R}\}$ such that $\bar{x}$ is the x-coordinate of one point in Q_0. We define a set Q_1 where for each placement $P_i' \in \mathcal{L}_i$ of each $P_i \in \mathcal{P}'$, each edge e of a triangle in the triangulation of P_i', and each vertical line $L \in \mathcal{V}$ we add to Q_1 the intersection of e and L. Additionally, we add to Q_1 the intersection of each line in $L \in \mathcal{V}$ with the two boundary edges of K. Let Q_2 denote the set of all intersections of pairs of line segments whose respective endpoints are in $Q_0 \cup Q_1$. We define $Q := Q_0 \cup Q_1 \cup Q_2$. A result in the work of Adamaszek and Wiese [1] implies that there exists a balanced cheap cut whose vertices are all contained in Q.

Lemma 43 ([1]). *Let $\epsilon > 0$ and let $\mathcal{P}'$ be a set of pairwise non-intersecting polygons in the plane with at most Λ edges each such that* $\text{area}(P) < \text{area}(\mathcal{P}')/3$ *for each $P \in \mathcal{P}$. Then, there exists a balanced $O(\epsilon\Lambda)$-cheap $\Lambda(\frac{1}{\epsilon})^{O(1)}$-cut Γ whose vertices are contained in Q.*

Our algorithm is recursive and places polygons from $\mathcal{P}'$, trying to maximize the total area of the placed polygons. In each recursive call, we are given an area $\bar{K} \subseteq K$ and a set of polygons $\bar{\mathcal{P}} \subseteq \mathcal{P}'$. In the main call, these parameters are $\bar{K} = K$ and $\bar{\mathcal{P}} = \mathcal{P}'$. If $\bar{\mathcal{P}} = \emptyset$, then we return an empty solution. If there is a polygon $P_i \in \bar{\mathcal{P}}$ with $\text{area}(P_i) \geq \text{area}(\bar{\mathcal{P}})/3$, then we guess a placement $P_i' \in \mathcal{L}_i$ and we recurse on the area $K \setminus P_i'$ and on the set $\bar{\mathcal{P}} \setminus \{P_i\}$. Otherwise, we guess the balanced cheap cut Γ due to Lemma 43 with $\epsilon := \frac{\delta}{\Lambda \log(n/\delta)}$, and for each shape $S \in \mathcal{S}$, we guess how many polygons of $\mathcal{P}'$ with shape S are contained in $\Gamma \cap \bar{K}$, how many are contained in $\bar{K} \setminus \Gamma$, and how many cross the boundary of Γ (i.e., have non-empty intersection with the boundary of Γ). Note that there are only $n^{(\Lambda \log(n/\delta))^{O(1)}}$ possibilities to enumerate. Let $\bar{\mathcal{P}}_{\text{in}}, \bar{\mathcal{P}}_{\text{out}}$, and $\bar{\mathcal{P}}_{\text{cross}}$ denote the respective sets of polygons. Then, we recurse on the area $\Gamma \cap \bar{K}$ with input polygons $\bar{\mathcal{P}}_{\text{in}}$ and on the area $\bar{K} \setminus \Gamma$ with input polygons $\bar{\mathcal{P}}_{\text{out}}$. Suppose that the recursive calls return two sets of polygons $\bar{\mathcal{P}}_{\text{in}}' \subseteq \bar{\mathcal{P}}_{\text{in}}$ and $\bar{\mathcal{P}}'_{\text{out}} \subseteq \bar{\mathcal{P}}_{\text{out}}$ that the algorithm managed to place inside the respective areas $\Gamma \cap \bar{K}$ and $\bar{K} \setminus \Gamma$. Then, we return the set $\bar{\mathcal{P}}_{\text{in}}' \cup \bar{\mathcal{P}}_{\text{out}}'$ for the guesses of $\Gamma, \bar{\mathcal{P}}_{\text{in}}, \bar{\mathcal{P}}_{\text{out}}$, and $\bar{\mathcal{P}}_{\text{cross}}$ that maximize $\text{area}(\bar{\mathcal{P}}_{\text{in}}' \cup \bar{\mathcal{P}}_{\text{out}}')$. If we guess the (correct) balanced cheap cut due to Lemma 43 in each iteration, then our recursion depth is $O(\log_{3/2}(n^2/\delta^2)) = O(\log(n/\delta))$ since the cuts are balanced and each polygon has an area of at least $\Omega(\text{area}(K)\delta^2/n^2)$ (see Lemma 40). Therefore, if in a recursive call of the algorithm the recursion depth is larger than $5\log(n/\delta) + 2$, then we return the empty set and do not recurse further. Additionally, if we guess the correct cut in each node of the recursion tree, then we cut polygons whose total area is at most a $\frac{\delta}{\log(n/\delta)}$-fraction of the area of all remaining polygons. Since our recursion depth is $O(\log(n/\delta))$, our algorithm outputs a packing for a set of polygons in $\mathcal{P}'$ with area at least $(1 - \frac{\delta}{\log(n/\delta)})^{O(\log(n/\delta))}\text{area}(\bar{\mathcal{P}}) = (1 - O(\delta))\text{area}(\bar{\mathcal{P}})$. We prove this formally in the following lemma.

Lemma 44. *Assume that there is a non-overlapping packing for $\mathcal{P}'$ in K. There is an algorithm with a running time of $n^{(\Lambda \log(n/\delta))^{O(1)}}$ that computes a placement of a set of polygons $\bar{\mathcal{P}}' \subseteq \mathcal{P}'$ inside K, increased by a factor $(1 + \delta)$ in both dimensions, such that* $\text{area}(\bar{\mathcal{P}}') \geq (1 - O(\delta))\text{area}(\mathcal{P}')$.

Proof. Recall that by Lemma 40, we can assume that the area of each polygon is at least $\frac{\delta^2}{2n^2}\text{area}(K)$ since we can pack polygons violating this in the extra capacity. We restrict our attention to the part of the recursion tree that corresponds to correct guesses (i.e., guesses corresponding to the optimal solution). This is justified, since in each node of the recursion tree, the algorithm selects the guess that maximizes the total area of the packed polygons.

In each node of the recursion tree, the input can be described by a region $\bar{K}$ and a set of polygons $\bar{\mathcal{P}}$ to be packed inside of $\bar{K}$. We prove the theorem by induction on the length of the longest path between each node to a leaf in the recursion tree. To be more specific, we show that if in a node $(\bar{K}, \bar{\mathcal{P}})$ the longest path between $(\bar{K}, \bar{\mathcal{P}})$ and a leaf is d, *and* there is a non-overlapping packing for $\bar{\mathcal{P}}$ in $\bar{K}$ in which each polygon $P_i \in \bar{\mathcal{P}}$ is placed on a position in the respective set $\mathcal{L}_i$, then the algorithm computes a solution which packs items whose total area is at least $(1 - \frac{\delta}{\log(n/\delta)})^d\text{area}(\bar{\mathcal{P}})$.

We begin by proving that the induction holds for the case $d = 0$. If $d = 0$, then the node $(\bar{K}, \bar{\mathcal{P}})$ is a leaf. If $\bar{\mathcal{P}} = \emptyset$, then the result trivially holds. Otherwise, the distance of $(\bar{K}, \bar{\mathcal{P}})$ to the root must be $5\log(n/\delta) + 2$, which we assume now. Since every guess on the path between the root and $(\bar{K}, \bar{\mathcal{P}})$ was correct, the area of the items that we need to pack was reduced by a factor of at least $2/3$ in each recursive step. Hence,

$$\begin{aligned}
\text{area}(\bar{\mathcal{P}}) &\le \left(\frac{2}{3}\right)^{5\log(n/\delta)+2} \text{area}(\mathcal{P}'),\\
&< \left(\frac{2}{3}\right)^{\frac{2}{\log(3/2)}\log\left(\frac{n}{\delta}\right)+\frac{\log(2)}{\log(3/2)}} \text{area}(K),\\
&= \left(\frac{2}{3}\right)^{\frac{\log\left(\frac{2n^2}{\delta^2}\right)}{\log\left(\frac{3}{2}\right)}} \text{area}(K),\\
&= \frac{\delta^2}{2n^2}\text{area}(K).
\end{aligned}$$

This implies that $\bar{\mathcal{P}} = \emptyset$, since by Lemma 40 each polygon in P' has area at least $\frac{\delta^2}{2n^2}\text{area}(K)$.

Suppose that $d > 0$; we distinguish the following two cases:

— *Case 1*: There exists a polygon $P_i' \in \bar{\mathcal{P}}$ such that $\text{area}(P_i) \ge \frac{1}{3}\text{area}(\bar{K})$. In this case, we guess the correct placement $P_i' \in \mathcal{L}_i$ for P_i and we recurse on the area $K \setminus P_i'$ and on the set $\bar{\mathcal{P}} \setminus \{P_i'\}$. By induction, the placed items have a total area of at least

$$\begin{aligned}
&\left(1-\frac{\delta}{\log(n/\delta)}\right)^{d-1} \text{area}(\bar{\mathcal{P}} \setminus \{P_i'\}) + \text{area}(P_i'),\\
&\ge \left(1-\frac{\delta}{\log(n/\delta)}\right)^{d} \text{area}(\bar{\mathcal{P}} \setminus \{P_i'\}) + \left(1-\frac{\delta}{\log(n/\delta)}\right)^{d} \text{area}(P_i'),\\
&= \left(1-\frac{\delta}{\log(n/\delta)}\right)^{d} \text{area}(\bar{\mathcal{P}}).
\end{aligned}$$

— *Case 2*: The second case is that every $P_i \in \bar{\mathcal{P}}$ satisfies that $\text{area}(P_i) \le \frac{1}{3}\text{area}(\bar{K})$. In this case, we enumerate all possible balanced cheap cuts Γ. For each enumerated cut Γ, we guess how many polygons of each shape are contained in $\Gamma \cap \bar{K}$ and in $\bar{K} \setminus \Gamma$, respectively. Let $\bar{\mathcal{P}}_{\text{in}}, \bar{\mathcal{P}}_{\text{out}}, \bar{\mathcal{P}}_{\text{cross}}$ be the associated set of polygons which lie inside, outside, or cross Γ. For each enumerated possibility of $\Gamma, \bar{\mathcal{P}}_{\text{in}}, \bar{\mathcal{P}}_{\text{out}}, \bar{\mathcal{P}}_{\text{cross}}$, we take the union of the packings corresponding to the nodes $(\Gamma \cap \bar{K}, \bar{\mathcal{P}}_{\text{in}})$ and $(\Gamma \setminus \bar{K}, \bar{\mathcal{P}}_{\text{out}})$ and finally choose the balanced cheap cut Γ and the sets $\bar{\mathcal{P}}_{\text{in}}, \bar{\mathcal{P}}_{\text{out}}, \bar{\mathcal{P}}_{\text{cross}}$ such that the area of packed polygons is maximized. We show that there exists such a packing with large area. We know that the elements of $\bar{\mathcal{P}}'$ can be packed into $\bar{K}$. Let Γ^* be the balanced cheap cut associated to this packing by Lemma 43. By induction on $(\Gamma^* \cap \bar{K}, \bar{\mathcal{P}}_{\text{in}})$, we get that the algorithm packs polygons from $\bar{\mathcal{P}}_{\text{in}}$ with a total area of at least $(1-\frac{\delta}{\log(n/\delta)})^{d-1}\text{area}(\bar{\mathcal{P}}_{\text{in}})$ into $(\Gamma^* \cap \bar{K})$. Similarly, the algorithm packs polygons from $\bar{\mathcal{P}}_{\text{out}}$ with a total area at least $(1-\frac{\delta}{\log(n/\delta)})^{d-1}\text{area}(\bar{\mathcal{P}}_{\text{out}})$ into $(\bar{K} \setminus \Gamma^*)$. Hence, the total area of the packed polygons is at least

$$\begin{aligned}
&\left(1-\frac{\delta}{\log(n/\delta)}\right)^{d-1} \text{area}(\bar{\mathcal{P}}_{\text{in}}) + \left(1-\frac{\delta}{\log(n/\delta)}\right)^{d-1} \text{area}(\bar{\mathcal{P}}_{\text{out}})\\
&= \left(1-\frac{\delta}{\log(n/\delta)}\right)^{d-1} (\text{area}(\bar{\mathcal{P}}_{\text{out}}) + \text{area}(\bar{\mathcal{P}}_{\text{in}}))\\
&= \left(1-\frac{\delta}{\log(n/\delta)}\right)^{d-1} (\text{area}(\bar{\mathcal{P}}) - \text{area}(\bar{\mathcal{P}}_{\text{cross}}))
\end{aligned}$$

$$\geq \left(1 - \frac{\delta}{\log(n/\delta)}\right)^d \text{area}(\bar{\mathcal{P}}).$$

Let $k := 5\log(n/\delta) + 2$ be the maximal recursion depth of the algorithm. We conclude the proof by using the induction above on $(K, \mathcal{P}')$ and noting that $k \leq 7\log(n/\delta)$ for $n \geq 2\delta$ (we find the optimal solution by complete enumeration for smaller n). We obtain that if there is a non-overlapping packing for $\mathcal{P}'$ in K, the algorithm computes a placement of a set of polygons $\bar{\mathcal{P}}'$ inside K of area at least

$$\begin{aligned}
&\left(1 - \frac{\delta}{\log(n/\delta)}\right)^k \text{area}(\mathcal{P}') \\
&\geq \left(1 - \frac{\delta}{\log(n/\delta)}\right)^{7\log(n/\delta)} \text{area}(\mathcal{P}') \\
&\geq \left(1 - \frac{\delta}{\log(n/\delta)} 7\log(n/\delta)\right) \text{area}(\mathcal{P}') \\
&= (1 - 7\delta)\text{area}(\mathcal{P}').
\end{aligned}$$

□

It remains to pack the polygons in $\tilde{\mathcal{P}}' := \mathcal{P}' \setminus \bar{\mathcal{P}}'$. The total area of their bounding boxes is bounded by $\sum_{P_i \in \tilde{\mathcal{P}}'} B_i \leq 2\text{area}(\tilde{\mathcal{P}}') = O(\delta)\text{area}(\mathcal{P}') = O(\delta)\text{area}(K)$. Therefore, we can pack them into additional space that we gain via increasing the size of K by a factor $1 + O(\delta)$, using the Next-Fit-Decreasing-Height algorithm [4]. This algorithm processes the rectangles by decreasing height and packs the rectangles in "shelves" starting from left to right on a "shelf" until there is no more space to pack the next rectangle. Then, it proceeds by defining the next "shelf" on top of the currently packed rectangles and resumes the packing on the new "shelf" from left to right.

This completes the description of our algorithm and its proof of correctness. In particular, this proves Theorem 4.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their numerous helpful comments which helped the presentation of the article a lot. In particular, the current proof of Lemma 5 was suggested by one the reviewers.

REFERENCES

[1] Anna Adamaszek and Andreas Wiese. 2014. A QPTAS for maximum weight independent set of polygons with polylogarithmically many vertices. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14)*. 645–656.

[2] Anna Adamaszek and Andreas Wiese. 2015. A quasi-PTAS for the two-dimensional geometric knapsack problem. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'15)*. 1491–1505.

[3] Nikhil Bansal, Alberto Caprara, Klaus Jansen, Lars Prädel, and Maxim Sviridenko. 2009. A structural lemma in 2-dimensional packing, and its implications on approximability. In *Algorithms and Computation*. Lecture Notes in Computer Science, Vol. 5878. Springer, 77–86.

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

[5] Waldo Gálvez, Fabrizio Grandoni, Sandy Heydrich, Salvatore Ingala, Arindam Khan, and Andreas Wiese. 2017. Approximating geometric knapsack via L-packings. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS'17)*. 260–271. https://doi.org/10.1109/FOCS.2017.32

[6] Sandy Heydrich and Andreas Wiese. 2019. Faster approximation schemes for the two-dimensional knapsack problem. *ACM Transactions on Algorithms* 15, 4 (2019), Article 47, 28 pages. https://doi.org/10.1145/3338512

[7] Klaus Jansen and Roberto Solis-Oba. 2007. New approximability results for 2-dimensional packing problems. In *Mathematical Foundations of Computer Science 2007*. Lecture Notes in Computer Science, Vol. 4708. Springer, 103–114.

[8] Klaus Jansen and Roberto Solis-Oba. 2008. A polynomial time approximation scheme for the square packing problem. In *Integer Programming and Combinatorial Optimization*. Lecture Notes in Computer Science, Vol. 5035. Springer, 184–198.
[9] Klaus Jansen and Guochuan Zhang. 2004. Maximizing the number of packed rectangles. In *Algorithm Theory—SWAT 2004*. Lecture Notes in Computer Science, Vol. 3111. Springer, 362–371.
[10] Klaus Jansen and Guochuan Zhang. 2004. On rectangle packing: Maximizing benefits. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'04)*. 204–213.
[11] Ce Jin. 2019. An improved FPTAS for 0-1 knapsack. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP'19)*. Article 76, 14 pages. https://doi.org/10.4230/LIPIcs.ICALP.2019.76
[12] Joseph Y. T. Leung, Tommy W. Tam, C. S. Wong, Gilbert H. Young, and Francis Y. L. Chin. 1990. Packing squares into a square. *Journal of Parallel and Distributed Computing* 10, 3 (1990), 271–275.
[13] Carla Negri Lintzmayer, Flávio Keidi Miyazawa, and Eduardo Candido Xavier. 2018. Two-dimensional knapsack for circles. In *Proceedings of the Latin American Symposium on Theoretical Informatics*. 741–754.
[14] A. Steinberg. 1997. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing* 26, 2 (1997), 401–409.
[15] V. V. Vazirani. 2001. *Approximation Algorithms*. Springer-Verlag, Berlin, Germany.

Received 10 September 2021; revised 3 August 2023; accepted 23 October 2023