Improved Approximations for Translational Packing of Convex Polygons

Adam Kurpisz
Bern University of Applied Sciences, Switzerland
Department of Mathematics, ETH Zürich, Switzerland

Silvan Suter
Department of Mathematics, ETH Zürich, Switzerland

Abstract

Optimal packing of objects in containers is a critical problem in various real-life and industrial applications. This paper investigates the two-dimensional packing of convex polygons without rotations, where only translations are allowed. We study different settings depending on the type of containers used, including minimizing the number of containers or the size of the container based on an objective function.

Building on prior research in the field, we develop polynomial-time algorithms with improved approximation guarantees upon the best-known results by Alt, de Berg and Knauer, as well as Aamand, Abrahamsen, Beretta and Kleist, for problems such as Polygon Area Minimization, Polygon Perimeter Minimization, Polygon Strip Packing, and Polygon Bin Packing. Our approach utilizes a sequence of object transformations that allows sorting by height and orientation, thus enhancing the effectiveness of shelf packing algorithms for polygon packing problems. In addition, we present efficient approximation algorithms for special cases of the Polygon Bin Packing problem, progressing toward solving an open question concerning an $\mathcal{O}(1)$-approximation algorithm for arbitrary polygons.

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

Keywords and phrases Approximation algorithms, Packing problems, Convex polygons, Bin packing, Strip packing, Area minimization

Digital Object Identifier 10.4230/LIPIcs.ESA.2023.76

Related Version *Full Version*: `https://arxiv.org/abs/2308.08523`

1 Introduction

Many real-life situations require us to make decisions about optimally packing a collection of objects into a specific container. One particular category of these packing problems is two-dimensional packing, which is encountered in everyday scenarios like arranging items on a shelf and in industrial applications such as cutting cookies from rolled-out dough or manufacturing sets of tiles from standard-sized panels made of wood, glass, or metal. Another intriguing example involves cutting fabric pieces for clothing production. In this case, the pieces often cannot be rotated freely, as they must adhere to a desired pattern in the final product, which is tailored of multiple elements. The widespread applicability of two-dimensional packing problems has led to a surge of interest in designing efficient algorithms to address them. In this paper, we follow the line of research and study the problem of packing convex polygons without rotations in various settings depending on the type of containers used.

Past research focusing on theoretical considerations of two-dimensional packings mainly concentrates on the scenario when all objects are axis-parallel rectangles. In this paper, we will discuss packing without rotations, in which only translations are permitted. There are two main classes of the problem depending on whether the size of the container is fixed and we want to minimize the number of containers used or whether we want to minimize the container's size with respect to some objective function.

31st Annual European Symposium on Algorithms (ESA 2023).
Editors: Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman; Article No. 76; pp. 76:1–76:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A seminal example of the first class is the Geometric Bin Packing problem in which a number of unit size squared bins to pack is to be minimized. The problem is arguably the most natural generalization of the regular ($1D$) Bin Packing to two dimensions, and its absolute approximability has been fully understood. Unless $\mathcal{P} = \mathcal{NP}$, the best possible efficient constant factor approximation is 2 [15], and such an algorithm is known [10].

In the second class, there are several variants to be considered. An example is the Strip Packing Problem which is concerned with packing objects into a strip of width 1 and infinite height in such a way that the maximum of all the heights of the placed objects is minimized. Like Geometric Bin Packing, Strip Packing generalizes ($1D$) Bin Packing. The best known efficient constant factor approximation for Strip Packing has approximation factor $(5/3 + \epsilon)$ [9]. It is known that there can not exist a polynomial time algorithm with an approximation ratio of $(3/2 - \epsilon)$ for any $\epsilon > 0$ unless $\mathcal{P} = \mathcal{NP}$, which follows directly from the approximation hardness of ($1D$) Bin Packing. Both classes of problems have been also considered in the asymptotic setting, see e.g. [11, 5, 12].

In younger time, there was also an increase of interest in cases where the objects in question are general convex polygons. Alt, de Berg and Knauer [2, 3] considered the problem of packing an instance consisting of a number of convex polygons of the form $p \subset [0, 1]^2$ into a minimum area axis-parallel rectangular container. We refer to this problem as Polygon Area Minimization throughout this paper. In the special case where the instance consists of rectangles only, the problem is known to admit a PTAS [4]. They proved the existence of the following efficient algorithm:

- A 23.78-approximation for Polygon Area Minimization.

Recently, Aamand, Abrahamsen, Beretta and Kleist [1] showed that the algorithm of Alt, de Berg and Knauer can be leveraged to obtain also efficient approximation algorithms of further problems:

- A 7-approximation for Polygon Perimeter Minimization.
- A 51-approximation for Polygon Strip Packing.
- An 11-approximation for Polygon Bin Packing for polygons with diameter at most $\frac{1}{10}$.

1.1 Our results

The results of Alt, de Berg and Knauer [2, 3] and Aamand et al. [1] are heavily based on so-called shelf packing algorithms. In shelf packing algorithms, the objects are first placed on the shelves, possibly ordered by height, which are later stacked on one another to build a final solution. Compared to axis-parallel rectangles, the main challenge in designing an approximation algorithm for polygon packing problems is that objects cannot be sorted by height and orientation simultaneously. As a result, the algorithm and its analysis in [2, 3] have such a large approximation guarantee. In this paper, we provide new insight into how shelf packing algorithms should be applied to polygon packing problems. We introduce a sequence of transformations of the objects that allow us first to sort them by height and later by orientation to build a solution of a much better approximation guarantee. More precisely, we design polynomial-time algorithms with the following factors:

- A 9.45-approximation for Polygon Area Minimization.

Using this algorithm as a subroutine, we build upon the methods from Aamand et al. [1] to obtain the following efficient approximation algorithms:

- A $(3.75 + \epsilon)$-approximation for Polygon Perimeter Minimization.
- A 21.89-approximation for Polygon Strip Packing.
- A 5.09-approximation for Polygon Bin Packing for polygons which have their diameter bounded by $\frac{1}{10}$.

The results are proved in Sections 4, 5, 6, and 7 respectively. Furthermore, concerning Polygon Bin Packing, in the full version of this paper we show the following results, which make progress towards solving an open question of a $\mathcal{O}(1)$-approximation algorithm for Polygon Bin Packing for arbitrary polygons.

- There is an efficient $\mathcal{O}(\frac{1}{\delta})$-approximation algorithm for Polygon Bin Packing for instances where each polygon has width or height at most $1-\delta$.
- There is an efficient $\mathcal{O}(1)$-approximation algorithm for Polygon Bin Packing for instances with the property that all polygons share a spine (up to translation) with height at least $\frac{3}{4}$.

2 Preliminaries

We start out considerations by recalling a classical and well-known problem in theoretical computer science, the Bin Packing problem: Given a list of numbers $s_1, \ldots, s_n \in (0,1] \cap \mathbb{Q}$, representing the sizes of n objects, the goal is to find the minimum number of bins of size 1, so that we can pack all objects into them. Bin Packing can be seen as the task of packing (1-dimensional) intervals into as few intervals of length 1 as possible. This definition can be extended to the two dimensional case. To do so we introduce several definitions.

Throughout the paper, for a set subset A of the domain of a function f, $f(A)$ is a shorthand notation for $\sum_{a \in A} f(a)$. A **packing instance** is defined by a finite set I containing objects to pack, a countable set $\mathcal{R}$ containing bins to pack the objects into and a set Φ of allowed transformations. A **shape** is a compact connected set $s \subset \mathbb{R}^2_{\geq 0}$. We denote a rectangular shape $r \subset \mathbb{R}^2_{\geq 0}$ by a tuple $r = (w,h) \in \mathbb{R}^2_{>0}$, which has width w and height h. An **object** o is an element of I and has a shape $s \subset [0,1]^2$. Furthermore, we define $\mathcal{I}_{\square}$ and $\mathcal{I}_{\pentagon}$ to be the sets which contain all finite sets I with objects consisting of axis-parallel rectangles and convex polygons, respectively. We will usually denote $|I|$ by n. In order to ensure computability, we assume that each object in $\mathcal{I}_{\square}$ and $\mathcal{I}_{\pentagon}$ is defined by finitely many vertices.

A **bin** $R \in \mathcal{R}$ is also characterized by having a shape. In this paper, we always assume that the shape of a bin R is an axis-parallel rectangle. That is, it is a rectangle with each of its sides being parallel to one of the primal axes in $\mathbb{R}^2$, and normally, its lower left corner is at the origin $(0,0) \in \mathbb{R}^2$. We write $\texttt{width}(R)$ and $\texttt{height}(R)$ for the width and the height of a bin $R \in \mathcal{R}$. If $R = [0,1]^2$, we say that R is a **unit bin**. In this study, the set of **allowed transformations** Φ is the set of all translations, i.e. $\Phi = \{\phi : R^2 \to \mathbb{R}^2 \mid \exists x_0 \in \mathbb{R}^2 \; \forall x \in \mathbb{R}^2 : \phi(x) = x + x_0\}$. Note that we consider the setting where rotations or reflections of objects are not allowed. We denote the width and the height of an object o (the length of the projection on the x-axis or y-axis respectively) by $\texttt{width}(o)$ and $\texttt{height}(o)$. The maximum width and height of a shape of an object in I, we denote by $w_{\max}(I)$ and $h_{\max}(I)$ respectively. We will also use the notation $\texttt{width}(S)$ and $\texttt{height}(S)$ for arbitrary subsets $S \subset \mathbb{R}^2$

A **packing** P of I is defined as a set of pairwise disjoint placements of all $o \in I$ with respect to the set of translations Φ. Given a bin $R \in \mathcal{R}$, we say that we can **pack $\boldsymbol{I}$ into $\boldsymbol{R}$** if there is a packing P of I so that $P \subset R$. In this case, we also say that P is a **packing of $\boldsymbol{I}$ into $\boldsymbol{R}$**. If we can pack the objects I into the bin R, we call R a **bounding box** of I. If we have an at most countable collection of bins $\mathcal{R} := \{R_j\}_{j \in J}$, we say that we can **pack $\boldsymbol{I}$ into $\boldsymbol{\mathcal{R}}$**, if there is a partition $I = \bigsqcup_{j \in J} I_j$ so that we can pack the objects I_j into R_j for all $j \in J$. In such case, if for all $j \in J$, P_j is a packing of I_j into R_j, we refer to $\mathcal{P} := \{P_j\}_j$ as a **multi-packing**. For any $j \in J$, we say that **$\boldsymbol{o}$ is in the packing $\boldsymbol{P_j}$**, if $o \in I_j$. We define the $\texttt{area}(P)$ to be the area of the smallest axis-parallel rectangle containing P.

The definition already demonstrates the more difficult nature of multi-dimensional packings compared to one dimensional packings. Even if the objects to pack are axis-parallel rectangles, it is no longer sufficient to consider in which bin to pack which rectangle, but the exact position in the bin also matters.

In this paper we consider problems consisting in packing convex polygonal shapes into axis-parallel rectangular bins under translational transformations. More precisely we consider the following problems.

▶ **Problem 1** (Polygon Packing).

Input: *Convex polygons $I \in \mathcal{I}_{\pentagon}$.*

Goals:

- *Bin Packing: Find the minimum number $B \in \mathbb{N}$ so that we can pack I into B unit bins.*
- *Strip Packing: Find the minimum height $H \in \mathbb{Q}_{>0}$ so that we can pack I into a bin of width 1 and height H.*
- *Area Minimization: Find a bounding box $R \in \mathbb{R}^2_{>0}$ of I so that $f(R) = \texttt{width}(R) \cdot \texttt{height}(R)$ is minimal.*
- *Perimeter Minimization: Find a bounding box $R \in \mathbb{R}^2_{>0}$ of I so that $f(R) = 2(\texttt{width}(R) + \texttt{height}(R))$ is minimal.*
- *Minimum Square: Find a bounding box $R \in \mathbb{R}^2_{>0}$ so that $f(R) = \max\{\texttt{width}(R), \texttt{height}(R)\}$ is minimal.*

Throughout the paper for the problems under consideration, we denote the optimal value of an instance $I \in \mathcal{I}_{\pentagon}$ by $\operatorname{opt}(I)$.

3 Shelf Packing Algorithms

Introducing well-known shelf-packing algorithms involves basic ($1D$)-Bin Packing algorithms, such as NextFit (NF), FirstFit (FF), and BestFit (BF). These place items $s_1, \dots, s_n$ into bins sequentially, with s_{i+1} placed according to specific rules. If no placement adheres to the rule, a new bin is opened. The rules differ for each algorithm.

- NF, places s_{i+1} into the most recently opened bin, if it has enough space.
- FF, places s_{i+1} in the earliest opened bin in which it fits.
- BF, places s_{i+1} in the bin with least free space among bins in which s_{i+1} fits.

NF and FF are often preprocessed by sorting the items in non-increasing size, which are called NextFitDecreasing (NFD) and FirstFitDecreasing (FFD). It is not hard to show that NF is a 2-approximation for the Bin Packing problem [14]. Moreover, NF packs it into at most $1 + 2\sum_{i=1}^{n} s_i$ bins. BF and FF both have an approximation ratio of 1.7 [7, 8], which are tight. Furthermore, as shown in [13], if $s_i \leq \frac{1}{m}$ for all $i \in [n]$ and some $m \geq 2$, FF packs this instance into at most $1 + (1 + \frac{1}{m})\sum_{i=1}^{n} s_i$ bins. [1]

Variants of NF, FF and BF exist for 2D rectangle packing problems, called NextFitDecreasingHeight (NFDH), FirstFitDecreasingHeight (FFDH), and BestFitDecreasingHeight (BFDH). These shelf-packing algorithms were introduced for Strip Packing, placing rectangles $r_1, \dots, r_n \in \mathcal{I}_{\square}$ into a bin $R = [0, 1] \times [0, \infty)$ with infinite height. The three algorithms order rectangles $r_1, \dots, r_n$ in non-increasing height and place them sequentially

[1] In fact they show that FF packs such instance in $2 + (1 + \frac{1}{m})\sum_{i=1}^{n} s_i$ bins. With a strategy analogous to the one from the proof of Theorem 3 in [6] in the two-dimensional case, however, one can show that an additive factor of 1 is sufficient.

into shelves in R. A shelf is a horizontal strip, and rectangles can open new shelves. A shelf-packing algorithm places r_{i+1} in an existing shelf according to a rule or opens a new shelf. The placement is bottom-left without intersecting other rectangles.

- NFDH places r_{i+1} in the most recently opened shelf, if it fits.
- FFDH places r_{i+1} in the lowest possible shelf which has enough space.
- BFDH, as FFDH, allows for placing rectangles in a lower shelf than the most recently opened one. Here, r_{i+1} is placed in the one which has minimal free horizontal space at the right end of the shelf among all shelves, while still having at least $\texttt{width}(r_{i+1})$ of it.

Since NFDH and FFDH are of particular interest in our paper, we present the following two absolute approximability results for axis-parallel rectangle Strip Packing problem.

▶ **Theorem 2** (Theorem 1 [6]). *Let $I \in \mathcal{I}_{\square}$. The packing P obtained by* NFDH satisfies

$$\texttt{height}(P) \leq h_{\max}(I) + 2\,\texttt{area}(I) \leq 3\,\mathrm{opt}(I).$$

▶ **Theorem 3** (Theorem 3 [6]). *Let $m \in \mathbb{N}$ and $I \in \mathcal{I}_{\square}$ be satisfying that $w_{\max}(I) \leq \frac{1}{m}$. The packing P of I obtained by* FFDH satisfies

$$\texttt{height}(P) \leq h_{\max}(I) + \left(1 + \frac{1}{m}\right)\texttt{area}(I) \leq \left(2 + \frac{1}{m}\right)\mathrm{opt}(I).$$

4 An Efficient 9.45-Approximation for Polygon Area Minimization and 7-Approximation when all Polygons are x-Parallelograms

In this section, we prove that there is an efficient $9.\overline{4}$-approximation algorithm for Polygon Area Minimization, which improves the previous best approximation factor for polynomial time algorithms of 23.78 by Alt et al. [2, 3]. Based on that result, we also show a 7-approximation in the special case when all polygons are x-parallelograms, a special type of parallelograms we will introduce in Definition 5.

4.1 An Efficient 9.45-Approximation for Polygon Area Minimization

To start the discussions, we introduce the following definition.

▶ **Definition 4.** *Let $p \subset [0,1]^2$ be a polygon. A* ***spine*** *s of p is a (straight) line segment connecting a point in $\operatorname{argmin}_{(x,y)\in p} y$ with a point in $\operatorname{argmax}_{(x,y)\in p} y$. We call the angle between the x-axis in increasing direction and s the* ***angle of s****. We say that s* ***is tilted to the right*** *or* ***leans to the right****, if this angle is in $(0, \frac{\pi}{2}]$ and* ***is tilted to the left*** *or* ***leans to the left****, if it is in $[\frac{\pi}{2}, \pi)$.*

Sometimes we also talk about "the" spine of a polygon, implicitly assuming that one has been fixed. The algorithm of Alt et al. is a shelf-packing algorithm and ordering polygons by the angle of their spines is a crucial step. Our algorithm shares these two characteristics. A key difference is that our algorithm first packs the polygons into parallelograms that have two of their sides parallel to the x-axis. We give such parallelograms their own definition:

▶ **Definition 5.** *An* ***x-parallelogram*** *is a parallelogram $q \subset \mathbb{R}^2$ that has two of its sides parallel to the x-axis. Of those two sides, we call the one with lower y-coordinate the* ***base*** *of q. We write $\texttt{base}(q)$ for the length of the base of q and $\texttt{wside}(q)$ for the width of one of the sides of q which is not parallel to the x-axis. Similar as to the definition for spines of* *polygons, we say that q **is tilted to the right** or **leans to the right**, if the angle between its right side and the increasing direction of the x-axis is in $(0, \frac{\pi}{2}]$ and **is tilted to the left** or **leans to the left**, if it is in $[\frac{\pi}{2}, \pi)$. We refer to this angle simply as **angle of q**.*

For packing polygons into x-parallelograms, we prove the following result, which is a refined version of the discussions of Aamand et al. [1] in Subsection 5.2.4 of their paper.

▶ **Lemma 6.** *Let p be a convex polygon. Then there exists an x-parallelogram q that contains p and satisfies*

(i) $\texttt{height}(q) = \texttt{height}(p)$

(ii) $\texttt{base}(q) \leq \texttt{width}(p)$

(iii) $\texttt{wside}(q) \leq \texttt{width}(p)$

(iv) $\texttt{area}(q) \leq 2\,\texttt{area}(p)$.

Proof. First, we construct a bounding x-parallelogram as the one that can be seen in Figure 1. Let l_b and l_t be lines parallel to the x-axis and tangent to p, touching the bottom of p and the top of p respectively. Choose $p_b \in p \cap l_b$ and $p_t \in p \cap l_t$ and define s to be the line connecting p_b and p_t. Note that s is a spine of p. Let s_l and s_r be tangent to p and parallel to s, lying on the left and on the right of p. Let $p_l \in p \cap s_l$ and $p_r \in p \cap s_r$. We now define q to be the set bounded by l_b, l_t, s_l and s_r. Note that q is an x-parallelogram that satisfies (i). As the left and right sides of q are just translations of s, it also satisfies (iii), because of course $s \subset p$ by convexity of p.

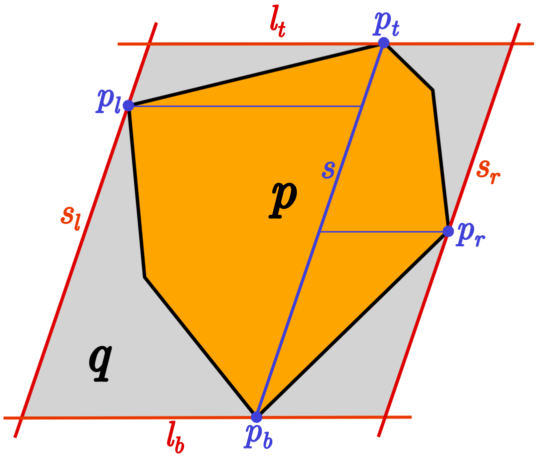

Figure 1 The construction of a bounding parallelogram q from the proof of Lemma 6 on an example polygon p. Note that here, it holds that $\texttt{base}(q) > \texttt{width}(p)$.

To see that q also satisfies (iv), we consider the triangle with vertices p_t, p_b and p_l. We note that it is contained in p due to convexity and contains exactly half the area of the part of q that lies on the left of s. Analogously, the triangle with vertices p_t, p_b and p_r has half the area of the part of q that lies on the right of s. So q does indeed also satisfy (iv).

However, q does not necessarily satisfy (ii) (see the polygon in Figure 1). If it does, then q satisfies all assumptions of the lemma and we are done.

So we consider now the case when $\texttt{base}(q) > \texttt{width}(p)$. Consider the axis-parallel rectangle r bounding p. That is, r contains p and each of its sides has non-empty intersection with p. Surely r satisfies $(i), (ii)$ and (iii). To see that it also satisfies (iv), note that

$$\begin{aligned}\texttt{area}(r) &= \texttt{width}(r)\,\texttt{height}(r)\\ &= \texttt{width}(p)\,\texttt{height}(p)\\ &< \texttt{base}(q)\,\texttt{height}(q)\\ &= \texttt{area}(q)\\ &\leq 2\,\texttt{area}(p).\end{aligned}$$

So whenever $\texttt{base}(q) > \texttt{base}(p)$, we showed that r satisfies all requirements $(i) - (iv)$ of the lemma instead. Since r is also an x-parallelogram, we conclude. ◀

With the help of Lemma 6, we can now present the main result of this chapter.

▶ **Theorem 7.** *There is a polynomial time $9.\overline{4}$-approximation algorithm for* Polygon Area Minimization*. Moreover, there is such algorithm with running time $\mathcal{O}(n^2 + N)$, where n is the number of polygons and N the total number of vertices in a given input $I \in \mathcal{I}_{\hexagon}$, assuming that each polygon $p \in I$ is given as a list of its vertices.*

Proof. Let $I \in \mathcal{I}_{\hexagon}$. We construct a packing P as the one depicted in Figure 2.

Pack each polygon $p \in I$ into an x-parallelogram q as in Lemma 6. Call the instance of all so-obtained x-parallelograms I_Q. The idea of our algorithm is to use FFDH to pack straightened, axis-parallel versions of the x-parallelograms I_Q and to then use this packing to obtain one for I_Q and hence also I which is not much bigger.

So, define yet another instance I_R which, for each $q \in I_Q$, contains a rectangle $r = (\texttt{base}(q), \texttt{height}(q))$. With FFDH, we now pack I_R into a strip of width $cw_{\max}(I)$, where $c \geq 1$ is to be determined later. Call the so-obtained packing P_R. By Theorem 3 it follows

$$\texttt{height}(P_R)(cw_{\max}(I)) \leq \left(1 + \frac{1}{m}\right)\texttt{area}(I_R) + ch_{\max}(I_R)w_{\max}(I), \tag{1}$$

where $m = \lfloor c \rfloor$, as $w_{\max}(I_R) = \max_{q \in I_Q} \texttt{base}(q) \leq w_{\max}(I)$ by Lemma 6.

Let $S \subset I_Q$ be the parallelograms corresponding to the rectangles in a certain shelf in the packing P_R. We can pack S into a new shelf of width $(c+2)w_{\max}(I)$ by first ordering the parallelograms S by decreasing angle. Indeed, if we, after this ordering, place them all next to each other in the shelf, we note that now all bases of the parallelograms are connected to each other and hence the overlap on either side is at most $\max_{q \in S} \texttt{wside}(q) \leq w_{\max}(I)$ by Lemma 6. Put all such shelves on top of each other and call the so-obtained packing P_Q. Note that P_Q has the same height as P_R, but $\frac{c+2}{c}$ times its width. Finally, we pack each polygon into its respective parallelogram in the packing P_Q. Call this packing P. Then

$$\begin{aligned}\texttt{area}(P) &\leq \texttt{area}(P_Q)\\ &\leq \frac{c+2}{c}\texttt{height}(P_R)(cw_{\max}(I))\\ &\leq \frac{c+2}{c}\left(1 + \frac{1}{m}\right)\texttt{area}(I_R) + (c+2)h_{\max}(I_R)w_{\max}(I)\\ &= \frac{c+2}{c}\frac{m+1}{m}\texttt{area}(I_Q) + (c+2)h_{\max}(I_Q)w_{\max}(I)\\ &\leq 2\frac{c+2}{c}\frac{m+1}{m}\texttt{area}(I) + (c+2)h_{\max}(I)w_{\max}(I)\\ &\leq \left(2\frac{c+2}{c}\frac{m+1}{m} + (c+2)\right)\text{opt}(I).\end{aligned}$$

One can check that the minimum is attained at $c = 3$, in which case we get a $9.\overline{4}$-approximation.

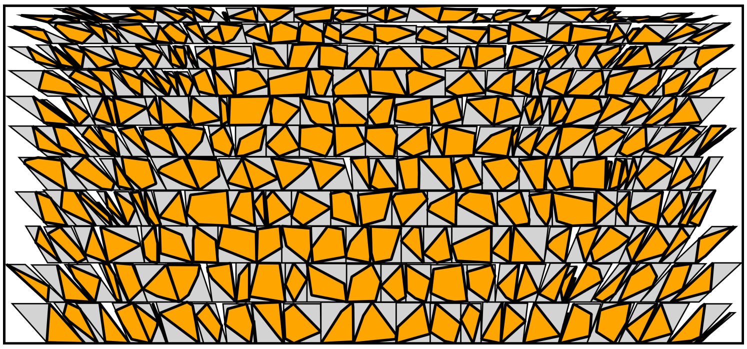

Figure 2 A packing computed with the algorithm from the proof of Theorem 7, here with parameter $c = 15$. For each polygon, also its computed bounding x-parallelogram is drawn.

We now note that the claimed running times of the above algorithm follows by observing that: Constructing I_Q from I can be done in $\mathcal{O}(N)$ time, constructing I_R from I_Q can be done in $\mathcal{O}(n)$ time, constructing P_R can be done in $\mathcal{O}(n^2)$ time, constructing P_Q from P_R can be done by sorting in $\mathcal{O}(n \log(n))$ time, and finally, constructing P from P_Q can be done in $\mathcal{O}(N)$ time. ◀

The algorithm of Alt et al. runs in time $\mathcal{O}(N \log(N))$ and thus for certain instances faster than our algorithm. If, in our algorithm, the use of FFDH for packing I_R is replaced by NFDH, one can show using Theorem 2 that for an optimal choice of $c = 2\sqrt{2}$, the algorithm has an approximation guarantee of 11.66 while having a running time of $\mathcal{O}(n \log(n) + N)$, thus obtaining an algorithm that runs faster than the algorithm of Alt et al., while still having a greatly improved approximation ratio.

4.2 An Efficient 7-Approximation for Polygon Area Minimization when all Polygons are x-Parallelograms

In the algorithm presented in the previous subsection, we first pack general polygons into x-parallelograms and afterwards pack these x-parallelograms. One would expect that if one wants to pack x-parallelograms from the start, one should be able to obtain a better approximation factor. Showing that this is indeed the case is the content of this brief section.

▶ **Theorem 8.** *There is a polynomial time 7-approximation algorithm for Polygon Area Minimization, when all input polygons are x-parallelograms.*

The proof follows along the lines of the proof of Theorem 7.

Proof. Let $I \in \mathcal{I}_{\lozenge}$ be so that every $p \in I$ is an x-parallelogram. As in the proof of Theorem 7, we define the set $I_R \in \mathcal{I}_{\square}$ that for every $p \in I$ contains some $r \in I_R$ with $\texttt{width}(r) = \texttt{base}(p)$ and $\texttt{height}(r) = \texttt{height}(p)$. Again, we pack I_R with FFDH into a strip of width $cw_{\max}(I)$ and call the so-obtained packing P_R.

After ordering them by angle, we can pack the parallelograms $S \subset I$ corresponding to the rectangles in some shelf in P_R into a new shelf of width $(c+2)w_{\max}(I)$, because of course $\texttt{wside}(p) \leq w_{\max}(I)$. Therefore, calling the so-obtained packing P,

$$\begin{aligned}\texttt{area}(P) &\leq \frac{c+2}{c}\texttt{area}(P_R)\\ &\leq \frac{c+2}{c}\frac{m+1}{m}\texttt{area}(I) + (c+2)h_{\max}(I)w_{\max}(I)\\ &\leq \left(\frac{c+2}{c}\frac{m+1}{m} + (c+2)\right)\operatorname{opt}(I),\end{aligned}$$

which, for $c = 2$, is minimized and equal to 7. ◀

5 An Efficient $(3.75 + \epsilon)$-Approximation for Polygon Perimeter Minimization and $(3.56 + \epsilon)$-Approximation for Polygon Minimum Square

In this section, we show how to leverage the algorithm from Theorem 7 to obtain an approximation-algorithm for Polygon Perimeter Minimization. We improve on the polynomial time 7.3-approximation algorithm obtained by Aamand et al. [1] and present an efficient $(3.75 + \epsilon)$-approximation. Moreover, we show how to leverage that result to obtain an $(3.56 + \epsilon)$-approximation-algorithm for Polygon Minimum Square.

5.1 An Efficient $(3.75 + \epsilon)$-Approximation for Polygon Perimeter Minimization

▶ **Theorem 9.** *For every $\epsilon > 0$, there is an efficient $(3.75 + \epsilon)$-approximation algorithm for Polygon Perimeter Minimization.*

Proof. Let $I \in \mathcal{I}_{\pentagon}$. Note that for the perimeter objective, it surely holds that

$$\operatorname{opt}(I) \geq 2(w_{\max}(I) + h_{\max}(I)). \tag{2}$$

Furthermore, since a bounding box of I has area at least $\texttt{area}(I)$ and the minimum perimeter rectangle having area $\texttt{area}(I)$ is a square, it also is true that

$$\operatorname{opt}(I) \geq 4\sqrt{\texttt{area}(I)}.$$

It follows from Inequality 2 that

$$\min\{h_{\max}(I), w_{\max}(I)\} \leq \frac{1}{4}\operatorname{opt}(I)$$

and without loss of generality, we assume that $w_{\max}(I) \leq \frac{1}{4}\operatorname{opt}(I)$. Otherwise we are making the following argument by packing into vertical shelves instead.

Let P be the packing obtained from the algorithm in Theorem 7, leaving c as a free parameter. Then P satisfies

$$\texttt{width}(P) \leq (c+2)w_{\max}(I)$$

and by dividing the inequality (1) by the width of the strip used for the rectangle packing obtained by FFDH, $cw_{\max}(I)$, we see that

$$\texttt{height}(P) \leq 2\frac{m+1}{m}\frac{\texttt{area}(I)}{cw_{\max}(I)} + h_{\max}(I) \leq \frac{1}{8}\frac{m+1}{m}\frac{\operatorname{opt}(I)^2}{cw_{\max}(I)} + h_{\max}(I).$$

We restrict the domain of c so that $c \geq \frac{\mathrm{opt}(I)}{4w_{\max}(I)} \geq 1$ and write $c = l\frac{\mathrm{opt}(I)}{4w_{\max}(I)}$ for some $l \geq 1$. Then

$$\texttt{width}(P) \leq \frac{l}{4}\,\mathrm{opt}(I) + 2w_{\max}(I), \quad \texttt{height}(P) \leq \frac{1}{2l}\frac{m+1}{m}\,\mathrm{opt}(I) + h_{\max}(I).$$

The perimeter of P is hence bounded by

$$\begin{aligned} 2(\texttt{width}(P) + \texttt{height}(P)) &\leq 2\left(\left(\frac{l}{4} + \frac{m+1}{2lm}\right)\mathrm{opt}(I) + 2w_{\max}(I) + h_{\max}(I)\right) \\ &\leq 2\left(\frac{1}{4}\left(l + \frac{2(m+1)}{lm}\right) + 1\right)\mathrm{opt}(I) \\ &= \frac{1}{2}\left(l + \frac{2(m+1)}{lm} + 4\right)\mathrm{opt}(I), \end{aligned}$$

which is minimized and equal to $3.75\,\mathrm{opt}(I)$ when l is equal to $\bar{l} := 2$, where we used that $m = \lfloor c \rfloor \geq \lfloor l \rfloor$.

Now since the value of $\mathrm{opt}(I)$ is not known beforehand and since $\bar{l}$ depends on it, we need to guess an optimal value for l. This can be done as follows. Compute the packing from the algorithm in Theorem 7 for $c = 1, (1+\epsilon), \ldots, (1+\epsilon)^K$, where $K = \frac{\log(n)}{\log(1+\epsilon)}$ and denote the packing obtained for $c = (1+\epsilon)^k$ by P_k for all $k \in \{0, 1, \ldots, n\}$. Over all those packings, choose the one that has minimum perimeter. Say this perimeter is $z > 0$. Let $k \in \{1, \ldots, K\}$ be so that

$$(1+\epsilon)^{k-1} \leq \bar{c} \leq (1+\epsilon)^k,$$

where $\bar{c} := \bar{l}\frac{\mathrm{opt}(I)}{4w_{\max}(I)}$. Then, for $l_k := (1+\epsilon)^k\frac{4w_{\max}(I)}{\mathrm{opt}(I)}$, it holds that

$$l_k = (1+\epsilon)^k\frac{4w_{\max}(I)}{\mathrm{opt}(I)} \leq (1+\epsilon)\bar{c}\frac{4w_{\max}(I)}{\mathrm{opt}(I)} = (1+\epsilon)\bar{l}.$$

In particular, as of course also $\bar{l} \leq l_k$, it holds that

$$\begin{aligned} z &\leq 2(\texttt{width}(P_k) + \texttt{height}(P_k)) \\ &\leq \frac{1}{2}\left(l_k + \frac{2(m+1)}{l_k m} + 4\right)\mathrm{opt}(I) \\ &\leq (1+\epsilon)\frac{1}{2}\left(\bar{l} + \frac{2(m+1)}{\bar{l}m} + 4\right)\mathrm{opt}(I) \\ &\leq (1+\epsilon)3.75\,\mathrm{opt}(I), \end{aligned}$$

which shows the statement. ◀

5.2 An Efficient $(3.56 + \epsilon)$-Approximation for Polygon Minimum Square

In this section, we show how to leverage the algorithm from Theorem 7 to obtain an approximation-algorithm for Polygon Minimum Square.

▶ **Theorem 10.** *For every $\epsilon > 0$, there is an efficient $(3.56 + \epsilon)$-approximation algorithm for Polygon Minimum Square.*

The proof is similar to the proof of Theorem 9.

Proof. Let $I \in \mathcal{I}_{\diamond}$. Note that

$$\mathrm{opt}(I) \geq \max\{w_{\max}(I), h_{\max}(I)\}$$

and also

$$\mathrm{opt}(I) \geq \sqrt{\mathtt{area}(I)}.$$

Analogously to the proof of Theorem 9, but substituting $c = l\frac{\mathrm{opt}(I)}{w_{\max}(I)}$ for some $l \geq 1$ instead, we can construct a packing P with

$$\mathtt{width}(P) \leq l\,\mathrm{opt}(I) + 2w_{\max}(I), \quad \mathtt{height}(P) \leq \frac{2}{l}\frac{m+1}{m}\mathrm{opt}(I) + h_{\max}(I).$$

Then

$$\max\{\mathtt{width}(P), \mathtt{height}(P)\} \leq \max\left\{l+2, \frac{2(m+1)}{lm} + 1\right\}\mathrm{opt}(I).$$

The minimum is attained for $l = \bar{l} := \frac{1}{2}(\sqrt{17}-1)$ in which case $m = 1$ and the approximation factor is equal to to $\frac{1}{2}(\sqrt{17}+3) \approx 3.56$. As in the proof of Theorem 9, we can guess the value of $\bar{l}$ to obtain a $(3.56+\epsilon)$-approximation algorithm. ◀

6 An Efficient 21.89-Approximation for Polygon Strip Packing

In this section, we show how to obtain a polynomial time $21.\overline{8}$-approximation for Polygon Strip Packing, improving on the previous best known approximation factor for efficient algorithms of 51 by Aamand et al. [1]. The idea is to construct vertical shelves with the algorithm from Theorem 7 and then to stack such vertical shelves horizontally into the strip. This idea comes from [1]. We slightly improve their procedure using the following observation.

▶ **Lemma 11.** *Let $\overline{I} \in \mathcal{I}_{\diamond}$ and let $\overline{P} \subset [0, (c+2)w_{\max}(\overline{I})] \times [0, \infty)$ be a shelf-packing obtained by the algorithm from Theorem 7 for some $c \geq 1$. Let $I \subset \overline{I}$ be the polygons in one of the shelves of $\overline{P}$ and define the packing $P := \{\overline{P}(p)\}_{p \in I}$. Then there is a shelf-packing P' of I into two shelves with $\mathtt{width}(P') \leq \frac{1}{2}(c+3)w_{\max}(\overline{I})$ and $\mathtt{height}(P') \leq 2\,\mathtt{height}(P)$.*

Proof. The idea is to simply split the shelf in half in its middle as follows. Let $x_{\mathrm{mid}} := \frac{(c+2)w_{\max}(\overline{I})}{2}$ and partition I into the two sets

$$I_L := \left\{p \in I \,|\, \mathtt{width}\left(P(p) \cap (\mathbb{R}_{\geq x_{\mathrm{mid}}} \times \mathbb{R})\right) \leq \frac{\mathtt{width}(p)}{2}\right\}$$

and $I_R = I \backslash I_L$. Note that

$$I_R \subset \left\{p \in I \,|\, \mathtt{width}\left(P(p) \cap (\mathbb{R}_{\leq x_{\mathrm{mid}}} \times \mathbb{R})\right) < \frac{\mathtt{width}(p)}{2}\right\}.$$

We can now pack I_L and I_R into separate shelves while respecting the ordering of the polygons in P. We denote the packing where both of those shelves are stacked on each other by P'. Both shelves have width at most

$$x_{\mathrm{mid}} + \frac{w_{\max}(I)}{2} \leq x_{\mathrm{mid}} + \frac{w_{\max}(\overline{I})}{2} = \frac{1}{2}(c+3)w_{\max}(\overline{I}).$$

and hence P' does as well. ◀

Making use of the lemma, we can show the following result.

▶ **Theorem 12.** *There is a polynomial time* $21.\overline{8}$*-approximation algorithm for* Polygon Strip Packing*.*

Proof. Let $I \in \mathcal{I}_{\diamond}$. We apply the algorithm from Theorem 7, for some c which we will fix later, to construct vertical shelves. More precisely, we rotate each item by an angle of $\pi/2$, apply the algorithm to the rotated instance, and then rotate the whole packing back by $\pi/2$. Let P be the so obtained packing and let $S_1, \dots, S_k$ be the vertical shelves of this packing.

We now stack $S_1, \dots, S_k$ horizontally into the (vertical) strip. That is, we pack the $\{S_i\}_{i\in[k]}$ into horizontal shelves $T_1, \dots, T_l$ themselves. Note that since each shelf S_i, where $i \in [k]$, has the same height, the problem reduces to ($1D$)-Bin Packing. In particular, stacking the $\{S_i\}_{i\in[k]}$ greedily, each except for possibly the last horizontal shelf is covered by at least half by the S_i. If the last shelf T_l is covered by half, we need at most $l \le \frac{2\,\texttt{area}(P)}{(c+2)h_{\max}(I)}$ horizontal shelves to pack $S_1, \dots, S_k$. Otherwise, we need at most $l \le \frac{2\,\texttt{area}(P)}{(c+2)h_{\max}(I)} + 1$ shelves. However, in this case we can reduce the height of T_l, which is filled less than half by the $\{S_i\}_{i\in[k]}$, to $\frac{1}{2}(c+3)h_{\max}(I)$ by using Lemma 11 on every shelf S in T_l.

In particular, for such packing P' it holds that

$$\begin{aligned}
\texttt{height}(P') &\le \left(\frac{2\,\texttt{area}(P)}{(c+2)h_{\max}(I)}\right)(c+2)h_{\max}(I) + \frac{1}{2}(c+3)h_{\max}(I) \\
&= 2\,\texttt{area}(P) + \frac{1}{2}(c+3)h_{\max}(I) \\
&\le \left(4\frac{c+2}{c}\frac{m+1}{m} + \frac{5}{2}c + \frac{11}{2}\right)\mathrm{opt}(I) \\
&= 21.\overline{8}\,\mathrm{opt}(I),
\end{aligned}$$

which is attained for $c = 3$. ◀

7 Efficient Approximation Algorithms for Polygon Bin Packing for Polygons of Width Upper Bounded by $1/M$

In this section, we present polynomial time approximation algorithms for Polygon Bin Packing for the case when polygons have their width bounded by a fraction of the form $\frac{1}{M}$ and also improved approximations if their height is bounded as well.

To the best of our knowledge, the only presently published polynomial time approximation algorithm for Polygon Bin Packing is an 11-approximation in the case when the diameter of the input polygons is bounded by $\frac{1}{10}$, see [1]. For this case, we obtain an efficient 5.09-approximation.

We prove the following theorem.

▶ **Theorem 13.** *Let $I \in \mathcal{I}_{\diamond}$ and assume that there is some $M \in \mathbb{N}_{\ge 2}$ with $w_{\max}(I) \le \frac{1}{M}$. Then one can pack I into*

$$\begin{cases} 32\,\mathrm{opt}(I) + 5 & \text{if } M = 2 \\ \frac{4M(M-1)}{(M-2)^2}\,\mathrm{opt}(I) + 3 & \text{if } M \ge 3 \end{cases}$$

bins efficiently. If additionally $h_{\max}(I) \le \frac{1}{M}$, then we can even efficiently pack I into

$$\begin{cases} 24\,\mathrm{opt}(I) + 3 & \text{if } M = 2 \\ \frac{2(M+1)(M-1)}{(M-2)^2}\,\mathrm{opt}(I) + 2 & \text{if } M \ge 3 \end{cases}$$

bins.

In particular, for inputs where both width and height of all polygons are upper bounded by $\frac{1}{10}$, we get a 5.09-approximation.

Proof. Assume first that $M \geq 3$. We use the algorithm from Theorem 7 for $c = \frac{1}{w_{\max}(I)} - 2 \geq M - 2$ to obtain a packing P with shelves $S_1, \ldots, S_k$. The packing P satisfies

$$\texttt{width}(P) \leq (c+2)w_{\max}(I) = 1,$$

and hence a shelf S_i, $i \in [k]$, fits into a unit bin. Note that $m = \lfloor c \rfloor \geq M - 2$, since $M - 2$ is an integer. Since $cw_{\max}(I) = 1 - 2w_{\max}(I) \geq \frac{M-2}{M}$, it holds that

$$\texttt{height}(P) \leq 2\left(1 + \frac{1}{m}\right)\frac{\texttt{area}(I)}{cw_{\max}(I)} + h_{\max}(I) \leq 2\left(1 + \frac{1}{M-2}\right)\frac{\texttt{area}(I)}{(M-2)/M} + h_{\max}(I).$$

We can now use NF to distribute the shelves into unit bins. Since $h_{max} = 1$, this uses at most $2\,\texttt{height}(P) + 1$ bins, see Section 3. But

$$2\,\texttt{height}(P) + 1 \leq 4\left(1 + \frac{1}{M-2}\right)\frac{\texttt{area}(I)}{(M-2)/M} + 2h_{\max}(I) + 1 \leq \frac{4M(M-1)}{(M-2)^2}\,\text{opt}(I) + 3.$$

If $h_{\max}(I) \leq \frac{1}{M}$ as well, we can use FF instead of NF to distribute the shelves into bins, which needs at most $(1 + \frac{1}{M})\,\texttt{height}(P) + 1$ bins, see again Section 3. This way, the number of needed bins is bounded by

$$\left(1 + \frac{1}{M}\right)\texttt{height}(P) + 1 \leq \frac{2(M+1)(M-1)}{(M-2)^2}\,\text{opt}(I) + 2,$$

as claimed.

Now we consider the case when $M = 2$. Partition I into two sets I_L and I_R, where a polygon $p \in I$ belongs to I_L if its chosen x-parallelogram in the proof of Theorem 7 is tilted to the left and to I_R if it is tilted to the right. We now proceed with the algorithm in the proof of Theorem 7 for $c = \frac{1}{w_{max}(I)} - 1 \geq 1$, but for I_L and I_R separately. Call the obtained packings P_L and P_R, respectively. Note that since all are tilted to one side only,

$$\texttt{width}(P_L),\, \texttt{width}(P_R) \leq (c+1)w_{\max}(I) = 1$$

and, using $m = |c| \geq 1$ and $cw_{\max}(I) = 1 - w_{\max}(I) \geq \frac{1}{2}$,

$$\texttt{height}(P_L) \leq 2\left(1 + \frac{1}{m}\right)\frac{\texttt{area}(I_L)}{cw_{\max}(I_L)} + h_{\max}(I_L) \leq 8\,\texttt{area}(I_L) + h_{\max}(I),$$

as well as analogously $\texttt{height}(P_R) \leq 8\,\texttt{area}(I_R) + h_{\max}(I)$.

Hence, for the packing P, where P_R is stacked on top of P_L, it holds that

$$\texttt{width}(P) \leq 1 \quad \text{and} \quad \texttt{height}(P) \leq 16\,\texttt{area}(I) + 2h_{\max}(I).$$

From here, with the same arguments as for $M \geq 3$, we obtain the desired bounds. ◀

References

1 Anders Aamand, Mikkel Abrahamsen, Lorenzo Beretta, and Linda Kleist. Online sorting and translational packing of convex polygons. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1806–1833, 2023. doi:10.1137/1.9781611977554.ch69.

2 Helmut Alt, Mark de Berg, and Christian Knauer. Approximating minimum-area rectangular and convex containers for packing convex polygons. *Journal of Computational Geometry*, 8(1):1–10, 2017. doi:10.20382/jocg.v8i1a1.

3 Helmut Alt, Mark de Berg, and Christian Knauer. Corrigendum to: Approximating minimum-area rectangular and convex containers for packing convex polygons. *Journal of Computational Geometry*, 11(1):653–655, 2020. doi:10.20382/jocg.v11i1a26.

4 Nikhil Bansal, José R. Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: Inapproximability results and approximation schemes. *Mathematics of Operations Research*, 31(1):31–49, 2006. doi:10.1287/moor.1050.0168.

5 Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In *Proceedings of the 2014 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 13–25, 2014. doi:10.1137/1.9781611973402.2.

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

7 György Dósa and Jiří Sgall. First Fit bin packing: A tight analysis. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, pages 538–549, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2013.538.

8 György Dósa and Jiří Sgall. Optimal analysis of best fit bin packing. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming*, pages 429–441, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-43948-7_36.

9 Rolf Harren, Klaus Jansen, Lars Prädel, and Rob van Stee. A $(5/3+\epsilon)$-approximation for strip packing. In Frank Dehne, John Iacono, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures*, pages 475–487, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. doi:10.1007/978-3-642-22300-6_40.

10 Rolf Harren and Rob van Stee. An absolute 2-approximation algorithm for two-dimensional bin packing, 2009. doi:10.48550/arXiv.0903.2265.

11 Rebecca Hoberg and Thomas Rothvoss. A logarithmic additive integrality gap for bin packing. In *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2616–2625, 2017. doi:10.1137/1.9781611974782.172.

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

13 D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3(4):299–325, 1974. doi:10.1137/0203025.

14 D.S. Johnson. *Near-optimal bin packing algorithms.* PhD thesis, Massachusetts Institute of Technology, 1973.

15 Joseph Y-T. Leung, Tommy W. Tam, C.S. Wong, Gilbert H. Young, and Francis Y.L. Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3):271–275, 1990. doi:10.1016/0743-7315(90)90019-L.