On Packing Squares with Resource Augmentation: Maximizing the Profit *

Aleksei V. Fishkin **Olga Gerber*** **Klaus Jansen*** **Roberto Solis-Oba+†**

Max Planck Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
avf@mpi-sb.mpg.de

Institute of Computer Science and Applied Mathematics*
University of Kiel
Olshausenstr. 40, 24118 Kiel, Germany
oge,kj@informatik.uni-kiel.de

Department of Computer Science+
The University of Western Ontario
London, ON, N6A 5B7, Canada
solis@csd.uwo.ca

Abstract

We consider the problem of packing squares with profits into a bounded square region so as to maximize their total profit. More specifically, given a set L of n squares with positive profits, it is required to pack a subset of them into a unit size square region $[0,1]\times[0,1]$ so that the total profit of the squares packed is maximized. For any given positive accuracy $\varepsilon > 0$, we present an algorithm that outputs a packing of a subset of L in the augmented square region $[1+\varepsilon]\times[1+\varepsilon]$ with profit value at least $(1-\varepsilon)\mathrm{OPT}(L)$, where $\mathrm{OPT}(L)$ is the maximum profit that can be achieved by packing a subset of L in the unit square. The running time of the algorithm is polynomial in n for fixed ε.

Keywords: Square packing, approximation algorithms..

1 Introduction

There has recently been increasing interest in solving a variety of 2-dimensional packing problems such as strip packing (Kenyon & Rémila 1996, Schiermeyer 1994, Steinberg 1997), 2-dimensional bin packing (Bansal & Sviridenko 2004, Caprara 2002, Chung, Garey & Johnson 1982, Seiden & van Stee 2003), and rectangle packing (Baker, Brown & Katseff 1981, Baker, Calderbank, Coffman & Lagarias 1983, Jansen & Zhang 2004*b*). These problems play an important role in a variety of applications in Computer Science and Operations Research, e.g. cutting stock, VLSI design, image processing, and multiprocessor scheduling, just to name a few.

In this paper we address the problem of packing squares with profits into a unit size square region so as to maximize the total profit of the packed squares. More precisely, we are given a set L of n squares S_i $(i=1,\ldots,n)$ with side lengths $s_i \in (0,1]$ and positive profits $p_i \in \mathbb{Z}_+$. For a subset $L' \subseteq L$, a *packing* of L' into the unit square is a positioning of the squares L' within the square $[0,1]\times[0,1]$ such that all the squares of L' have disjoint interiors. The goal is to find a subset $L' \subseteq L$, and its packing into the unit square, of maximum profit, $\sum_{S_i\in L'} p_i$.

*Supported by EU-Project CRESCCO "Critical Resource Sharing for Cooperation in Complex Systems", IST-2001-33135 and by DFG-Graduiertenkolleg 357, Effiziente Algorithmen und Mehrskalenmethoden.

†Author partially supported by the Natural Sciences and Engineering Research Council of Canada grant R3050A01.

This problem is known to be strongly NP-hard even for the case of squares with identical profits (Leung, Tam, Wong, Young & Chin 1990). Hence, it is very unlikely that any polynomial time algorithm for this problem exists, and so, we look for efficient heuristics with good performance guarantees. A polynomial time algorithm A is said to be a ρ-*approximation algorithm* for a maximization problem Π if on every instance I of Π algorithm A outputs a feasible solution with a value $A(I) \geq \frac{1}{\rho}\cdot\mathrm{OPT}(I)$, where $\mathrm{OPT}(I)$ is the optimum. The value of $\rho \geq 1$ is called the *approximation ratio* or *performance guarantee*. A *polynomial time approximation scheme* (PTAS) is a family of approximation algorithms $\{A_\varepsilon\}_{\varepsilon>0}$ such that A_ε is a $(1+\varepsilon)$-approximation algorithm and its running time is polynomial in n for any fixed value $\varepsilon > 0$. If the running time of each A_ε is polynomial in the size of the instance and in $1/\varepsilon$, then $\{A_\varepsilon\}_{\varepsilon>0}$ is called a *fully polynomial time approximation scheme* (FPTAS).

Related results. The 1-dimensional version of the above packing problem is equivalent to the knapsack problem: given a knapsack of capacity B and a set of items with profits and sizes, pack items of total size at most B into the knapsack so that the total profit of the packed items is maximized. It is well-known that the knapsack problem is weakly NP-hard (Garey & Johnson 1979), and it admits a FPTAS (Kellerer, Pferschy & Pisinger 2004, Lawler 1979). In contrast, our problem is strongly NP-hard, and, hence, it admits no FPTAS unless P = NP.

In the 2-dimensional version of the problem, one can see a strong relationship to the the problem of packing squares into a rectangle of minimum area (Moser 1965, Moser & Pach 1989): given a set L of squares of total area at most 1 pack them into a rectangle of area x such that x is minimized. Regarding lower bounds for this latter problem, there is just one non-trivial result known (Novotny 1995): the set L of four squares with sides $s_1 = \sqrt{\frac{1}{2}}$, $s_2 = s_3 = s_4 = \sqrt{\frac{1}{6}}$ shows that the value of x is at least $\frac{2+\sqrt{3}}{3} > 1.244$. On the other hand, there are a number of quite complicated results yielding several upper bounds for this problem. As it was shown in (Moon & Moser 1967), any set L of squares with side lengths at most $s_{\max}$ can be packed into a square of size $a = s_{\max} + \sqrt{1-s_{\max}}$. Later in (Meir & Moser 1968), this result was extended by showing that any set L of squares of total area V can be packed into a rectangle of size $a_1 \times a_2$, provided that $a_1 > s_{\max}$, $a_2 > s_{\max}$ and $s_{\max}^2 + (a_1 - s_{\max})(a_2 - s_{\max}) \geq V$. Hence, the value of x is upper bounded by 2. Some further results in this direction were obtained in (Kleitman & Krieger 1975). In particular, it was proven that any set L of squares of total area V can be packed into a rectangle of size $\sqrt{2V} \times 2\sqrt{V}/\sqrt{3}$. Thus, substituting $V = 1$, the value of x is upper bounded by $\sqrt{\frac{8}{3}} \doteq 1.633$. Finally, the result presented in (Novotny 1996) shows that any set L of squares of total area 1 can be packed into a rectangle whose area is less than 1.53. Beyond the intrinsic interest of these results, the proofs – being constructive – provide algorithms for actually effecting the asserted packings.

There is also a relationship to the 2-dimensional bin packing problem: given a set L of rectangles of specified size (width, height), pack the rectangles into N square bins of unit area such that N is minimized. The problem is strongly NP-hard (Leung et al. 1990) and there is no better than a 2-approximation algorithm for it (Ferreira, Miyazawa & Wakabayashi 1999), unless P = NP. A long history of approximation results exists for this problem and its special cases (Bansal & Sviridenko 2004, Caprara 2002, Chung et al. 1982, Seiden & van Stee 2003). Very recently a number of best possible asymptotic results have been obtained (i.e. for the case when the optimum uses a large number of bins). In (Bansal & Sviridenko 2004) it was proven that the general version of the problem does not admit an asymptotic FPTAS, unless P = NP. However, there is a FPTAS if all rectangles are actually squares. Also, in (Correa & Kenyon 2004) a polynomial algorithm was presented which packs any set L of rectangles into at most $N^{opt}(L)$ augmented bins whose sides have length $(1+\varepsilon)$, where $N^{opt}(L)$ denotes the minimum number of bins required to pack the rectangles in L, and $\varepsilon > 0$. Interestingly, this algorithm can be adapted to produce a PTAS for the following very related problem: given a set L of rectangles of total area at most 1, pack them into a square of area x such that x is minimized. For a fixed target area x, the algorithm decides whether a packing of L into a square of area $(1+\varepsilon)x$ exists.

Finally, one can also find a relationship to strip packing (Gilmore & Gomory 1965): given a set L of rectangles it is required to pack the rectangles into a vertical strip $[0,1] \times [0,+\infty)$ so that the height of the packing is minimized. The strip packing problem is strongly NP-hard since it includes the classical bin packing problem as a special case. Many known strip packing ideas come from bin packing. The "Bottom-Left" heuristic has asymptotic performance ratio 2 when the rectangles are sorted by decreasing widths (Baker, Coffman & Rivest 1980). In (Coffman, Garey, Johnson & Tarjan 1980) several simple algorithms were studied where the rectangles are placed on "shelves" using one-dimensional bin-packing heuristics. It was shown that the First-Fit shelf algorithm has asymptotic performance ratio 1.7 when the rectangles are sorted by decreasing height (this defines the First-Fit-Decreasing-Height algorithm). The asymptotic performance ratio of the best heuristic was further reduced to $3/2$ (Sleator 1980), then to $4/3$ (Golan 1981), and to $5/4$ (Baker et al. 1981). Finally, in (Kenyon & Rémila 1996) it was shown that there exists an asymptotic FPTAS for the case when the sides of all rectangles in the set are at most 1. For the case of absolute performance ratio, the two currently best algorithms have the same performance ratio, 2 (Schiermeyer 1994, Steinberg 1997).

In contrast to all above mentioned problems, there are very few results known for packing rectangles into a rectangular region so as to maximize their total profit. For a long time the only known result was an asymptotic $(4/3)$-approximation algorithm for packing squares with unit profits into a rectangle (Baker et al. 1983). Only very recently this algorithm for packing unit profit squares has been improved to a PTAS (Jansen & Zhang 2004*a*). For packing rectangles, several approximability results were presented in (Jansen & Zhang 2004*b*). The best one is a $(2+\varepsilon)$-approximation algorithm, for any fixed $\varepsilon > 0$.

Naive approach. There is a natural two-step approach that could be used for our problem: first, use a knapsack FPTAS with accuracy $\delta \in (0,\varepsilon]$ to find a set L' of squares of total area at most 1 and maximum profit, and then apply one of the above algorithms to pack these squares inside a square region of minimum area. By the above discussion, this approach, yields a solution of total profit at least $(1-\varepsilon)$ times the optimum, but depending on the running time, the packing needs an augmented square region of size $\sqrt{2}$ or $x(1+\varepsilon)$, where x is the size of the minimum square region needed to pack L'.

This approach approximates the optimum profit quite well. However, the approach fails in the sense that the augmented square cannot be of size arbitrarily close to the unit one, since the value of x could be much larger than 1. Let $\varepsilon \in (0,1]$, and L be a set consisting of two large squares S_1, S_2 with side lengths $s_1, s_2 = 1/\sqrt{2}$ and profits $p_1 = p - \varepsilon p, p_2 = \varepsilon p$, and n^2 small squares S_i ($i = 3, \ldots, n^2+2$) with side lengths $s_i = 1/(\sqrt{2}n)$ and profits $p_i = \varepsilon p/n^2$, for some positive value p. For all small squares, their total area is

$$\sum_{i=3}^{n^2+2} (s_i)^2 = n^2/(2n^2) = \frac{1}{2}$$

and their total profit is

$$\sum_{i=3}^{n^2+2} p_i = n^2 \cdot (\varepsilon p/n^2) = \varepsilon p.$$

The corresponding knapsack problem for this set of squares can be formulated as:

$$\begin{array}{lll} \text{Maximize} & \sum_{i=1}^{n^2+2} p_i x_i & \\ \text{subject to} & \sum_{i=1}^{n^2+2} (s_i)^2 x_i & \leq 1, \\ & x_i & \in \{0,1\} \\ & \text{for all } i = 1, \ldots, n^2+2. & \end{array}$$

There are two optimum solutions for this knapsack problem:

(a) the two large squares S_1, S_2 are chosen; their area is $(s_1)^2 + (s_2)^2 = 1$ and their profit is $(p_1 + p_2) = p - \varepsilon p + \varepsilon p = p$, and

(b) the large square S_1 and all the small squares S_i ($i = 3, \ldots, n^2+2$) are chosen; their area is $\sum_{i=3}^{n^2+2} (s_i)^2 + (s_1)^2 = \frac{1}{2} + \frac{1}{(\sqrt{2})^2} = 1$ and their profit is $\sum_{i=3}^{n^2+2} p_i + p_1 = p - \varepsilon p + \varepsilon p = p$.

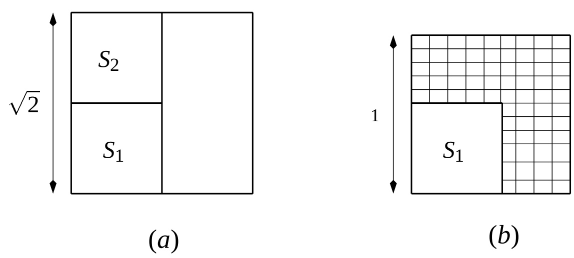

Figure 1: Example

If we use a FPTAS for the knapsack problem with accuracy $\delta \leq \varepsilon/2$, there is no guarantee that a solution of the form (b) is produced. If solution (a) is obtained, then its two large squares can only be packed into a square of side length $\sqrt{2}$. This is a large augmentation of the unit square, see Fig. 1. Hence, by using this naive approach we cannot guarantee that the augmented square has size arbitrarily close to 1.

Our results. Here we consider the so-called resource augmentation version of the square packing problem, that is, we allow the length of the unit square region where the squares are to be packed to be increased by some small value.Our main result is this:

Theorem 1.1. *For any set L of n squares and any fixed value $\varepsilon > 0$, there exists an algorithm A_ε which finds a subset of L and its packing into the augmented unit square $[0,1+\varepsilon]\times[0,1+\varepsilon]$ with profit*

$$A_\varepsilon(L) \geq (1-\varepsilon)\mathrm{OPT}(L),$$

where $\mathrm{OPT}(L)$ is the maximum profit that can be achieved by packing any subset of L in the original unit square region $[0,1]\times[0,1]$. The running time of A_ε is

$$O\left(\frac{n^2}{\varepsilon^2}\log(n/\varepsilon)\left(\frac{n}{\varepsilon^8\Delta^2}\right)^{1/\Delta^2}\right),$$

where $\Delta = \varepsilon^{4^{1/\varepsilon^2}}$.

Even though, the running time of algorithm A_ε is polynomial in n for fixed ε, it is exponential in $1/\varepsilon$. Therefore, our result is primarily of theoretical importance.

Our algorithm combines and refines several known approximation techniques used for knapsack problems, strip packing, and, scheduling problems. Contrasting to recent algorithms in (Bansal & Sviridenko 2004, Correa & Kenyon 2004, Jansen & Zhang 2004*b*, Kenyon & Rémila 1996), we use no LP formulations. Our algorithm is based on a few simple ingredients, which are quite easy to follow. This demonstrates that applying the resource augmentation technique to 2-dimensional packing problems may significantly simplify the task of designing approximation algorithms for them.

Our approach. First, we use a nice property of the problem. Namely, if all the squares in L are small, that is, their side lengths are at most ε^2, then we can simply find a knapsack solution, and then apply the Next-Fit-Decreasing-Height (NFDH) heuristic (see Section 2.2). This gives a packing in the augmented square $[0,1+\varepsilon]\times[0,1+\varepsilon]$ whose profit is at least $(1-\varepsilon)\mathrm{OPT}(L)$.

Hence, we use the idea of dealing separately with small and large squares. We partition the set L into groups: $L^{(0)}$ contains squares with side lengths in $(\varepsilon^4,1]$, and $L^{(j)}$ contains squares with side lengths in $(\varepsilon^{4^{j+1}},\varepsilon^{4^j}]$, for $j\geq 1$. Then, there must exist a group $L^{(k)}$, $0\leq k\leq 1/\varepsilon^2-1$, such that its contribution to the optimum solution is at most $\varepsilon^2\mathrm{OPT}(L)$. We simply drop the squares of $L^{(k)}$ from consideration. This causes a loss of at most a factor of ε^2 in the optimum. Then, we partition the squares $L\setminus L^{(k)}$ into large ones, $\cup_{j\leq k-1}L^{(j)}$, and small ones, $\cup_{j\geq k+1}L^{(j)}$.

If the set of large squares is not empty, we try to find the best possible selection of them to be included in our solution. Notice that each large square has side length at least $\varepsilon^{4^{1/\varepsilon^2-1}}$. Hence, there are at most $O(1)$ large squares in a valid packing. We augment the unit square to $[0,1+\varepsilon]\times[0,1+\varepsilon]$, and discretize the set of possible positions for the large squares in a packing. This allows us to enumerate all possible packings of large squares in $[0,1+\varepsilon]\times[0,1+\varepsilon]$. In each packing we try to fill the empty space with small squares. To do this we solve a knapsack problem and use a modified NFDH heuristic. Among all packings found we select the one with the maximum profit, which must be at least $(1-\varepsilon)\mathrm{OPT}(L)$.

Last notes. In the following sections we give our proof of Theorem 1.1, describing our approximation algorithm. Section 2 gives background and notation, as well as some preliminary results. Sections 3 and 4 give detailed explanations of the different steps of the algorithm. In Section 5 we give a description of the overall algorithm.

2 Preliminaries

For a subset of squares $L'\subseteq L$, we use $profit(L')$ and $size(L')$ to denote the profit, $\sum_{S_i\in L'}p_i$, and size, $\sum_{S_i\in L'}s_i\cdot s_i$, of L'. In addition, we use L^{opt} to denote an optimal subset of L that can be packed in the unit square $[0,1]\times[0,1]$. So,

$$profit(L^{opt}) = \mathrm{OPT}(L) \text{ and } size(L^{opt})\leq 1.$$

Throughout the paper we also assume that $\varepsilon\in(0,1/4)$ and the value of $1/\varepsilon$ is integral.

2.1 Solving the Knapsack Problem

Let L be a set of squares. We consider the following version of the knapsack problem. We are given a knapsack of capacity S and a set of squares $LIST\subseteq L$, where each square $S_i\in LIST$ has a size $(s_i)^2$ and profit p_i. It is required to find a subset $LIST(S)\subset LIST$ of maximum profit $\sum_{i\in LIST(S)}p_i$, given that $\sum_{S_i\in LIST(S)}(s_i)^2\leq S$, i.e., $LIST(S)$ fits in a knapsack of size S.

This knapsack problem is NP-hard, but it admits a FPTAS (Garey & Johnson 1979, Kellerer et al. 2004, Lawler 1979). Given a precision $\varepsilon'>0$, the FPTAS outputs a subset $LIST(S)\subseteq LIST$ such that

$$\begin{aligned} size(LIST(S)) &\leq S \\ profit(LIST(S)) &\geq (1-\varepsilon')\mathrm{OPT}(LIST,S), \end{aligned} \tag{1}$$

where $\mathrm{OPT}(LIST,S)$ is the maximum profit of $LIST$ with respect to the knapsack's capacity S. The running time of the FPTAS is polynomial in n and $1/\varepsilon'$.

2.2 The NFDH Heuristic

We consider the following simplified version of the square packing problem: given a subset $L'\subseteq L$ of squares with side lengths at most ε^2, and a rectangle $[0,a]\times[0,b]$ ($a,b\in[0,1]$) such that $size(L')\leq a\cdot b$, pack the squares of L' into the augmented rectangle $[0,a+\varepsilon^2]\times[0,b+\varepsilon^2]$.

First, we order the squares of L' decreasingly by side lengths. Then, we put the squares into the rectangle $[0,a]\times[0,b]$ by using the Next-Fit-Decreasing-Height (NFDH) heuristic; this packs the squares into a sequence of sublevels. The first sublevel is the bottom of the rectangle. Each subsequent sublevel is defined by a horizontal line drawn through the top of the largest square placed on the previous sublevel. Squares are packed in a left-justified manner, until there is some space to the vertical line at point a, i.e. the last square can be packed on this sublevel even if it intersects the right border of the rectangle $[0,a]\times[0,b]$. At that moment, the current sublevel is closed, a new sublevel is started and packing proceeds as above. For an illustration see Fig. 2.

We will use the following simple result.

Lemma 2.1. *Let $L'\subseteq L$ be any subset of squares with side lengths at most ε^2, ordered by non-increasing side lengths, and let $[0,a]\times[0,b]$ ($a,b\in[0,1]$) be a rectangle such that $size(L')\leq a\cdot b$. Then, the NFDH heuristic outputs a packing of L' in the augmented rectangle $[0,a+\varepsilon^2)\times[0,b+\varepsilon^2]$.*

Proof. Let q be the number of sublevels. Let h_i be the height of the first square on the ith sublevel. Recall that NFDH packs the squares of L' on sublevels in order of

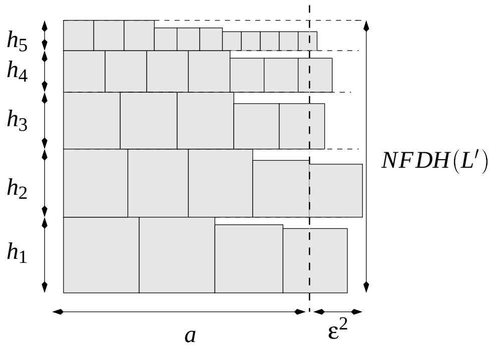

Figure 2: NFDH for small squares

non-increasing side lengths. Hence, the height of the packing is

$$H = \sum_{i=1}^{q} h_i.$$

Since the side of any square is at most ε^2, then

$$\varepsilon^2 \geq h_1 \geq h_2 \geq \ldots \geq h_q > 0.$$

Furthermore, the total width of the squares on each full sublevel is at least a and at most $a + \varepsilon^2$.

Since the squares L' are packed non-increasingly by side lengths, the total area of the squares on each ith ($i = 1, \ldots, q-1$) sublevel is at least $h_{i+1} \cdot a$. Assume that the value of H is larger than $b + \varepsilon^2$. Then, the area covered by squares would be at least

$$\begin{aligned}
&\sum_{i=1}^{q-1} h_{i+1} \cdot a = a \cdot \sum_{i=2}^{q} h_i \\
&= a \cdot [H - h_1] > a \cdot [(b + \varepsilon^2) - h_1] \ \text{ since } H > b + \varepsilon^2 \\
&= a \cdot [b + (\varepsilon^2 - h_1)] \geq a \cdot b = size(L') \text{ since } \ h_1 \leq \varepsilon^2.
\end{aligned}$$

That gives a contradiction.

Overall, the width of each sublevel is at most $a + \varepsilon^2$, and the height of the packing is at most $b + \varepsilon^2$. Hence, we have a packing in the augmented rectangle $[0, a+\varepsilon^2] \times [0, b+\varepsilon^2]$, and the lemma follows. ◻

Collorary 2.2. *If all squares in L have side length at most ε^2, then there is an algorithm which finds a subset of L and its packing in the augmented square $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$ with profit at least $(1-\varepsilon)$OPT. The running time of the algorithm is polynomial in n and $1/\varepsilon$.*

Proof. By solving a knapsack problem we can find a subset of L, whose total area is at most 1 and whose profit is at least $(1-\varepsilon)$OPT. By using NFDH we pack this subset into the augmented square $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$. ◻

2.3 Partitioning the Squares

Let L be a set of squares. We define the group $L^{(0)}$ of squares with side lengths in $(\varepsilon^4, 1]$, and for $j \in \mathbb{Z}_+$ we define the group $L^{(j)}$ of squares with side lengths in $(\varepsilon^{4^{j+1}}, \varepsilon^{4^j}]$. Then,

$$\cup_{j=0}^{\infty} L^{(j)} = L \text{ and } L^{(\ell)} \cap L^{(j)} = \emptyset, \text{ for } \ell \neq j.$$

We will use the following simple result.

Lemma 2.3. *There is a group $L^{(k)}$ with $0 \leq k \leq 1/\varepsilon^2 - 1$ such that its contribution to the optimum is*

$$profit(L^{opt} \cap L^{(k)}) \leq \varepsilon^2 \text{OPT},$$

where L^{opt} is an optimal subset of squares.

Proof. If there is one empty group L^k with $0 \leq k \leq 1/\varepsilon^2 - 1$, then its profit is zero and the result of lemma follows. So, assume that all groups $L^{(0)}, L^{(1)}, \ldots, L^{(1/\varepsilon^2-1)}$ are non-empty. Consider the optimal set $L^{opt} \subseteq L$ of squares. Recall that $L^{(\ell)} \cap L^{(j)} = \emptyset$ for all $\ell \neq j$. So,

$$\text{OPT} = profit(L^{opt}) \geq \sum_{j=0}^{1/\varepsilon^2-1} profit(L^{opt} \cap L^{(j)}).$$

Then, there must exist at least one group $L^{(k)}$ with $0 \leq k \leq 1/\varepsilon^2 - 1$ such that its contribution to the profit of the optimal solution is at most the average contribution of the $1/\varepsilon^2$ groups. Hence,

$$\begin{aligned}
profit(L^{(k)} \cap L^{opt}) &\leq [\sum_{j=0}^{1/\varepsilon^2-1} profit(L^{opt} \cap L^{(j)})]/(1/\varepsilon^2) \\
&\leq \varepsilon^2 \text{OPT}.
\end{aligned}$$

The lemma follows. ◻

Our idea is to drop the squares $L^{(k)}$ from consideration. Then, an optimal packing for $L \setminus L^{(k)}$ has profit at least $(1-\varepsilon^2)$OPT, i.e. this makes a loss of at most a factor of ε^2 in the optimum.

We separate the squares in $L \setminus L^{(k)}$ into two groups: $\cup_{j \leq k-1} L^{(j)}$ and $\cup_{j \geq k+1} L^{(j)}$. These groups are denoted as L_{large} and L_{small}. The squares in L_{large} and L_{small} are called large and small, respectively.

Collorary 2.4. *Let $\Delta = \varepsilon^{4^k}$, where k is as defined above. Then the side length of any large square is at least Δ and the side length of any small square is at most $\varepsilon^4 \Delta$. Moreover,*

$$profit(L^{opt} \cap [L_{large} \cup L_{small}]) \geq (1-\varepsilon^2)\text{OPT}.$$

3 Large Squares

Here we consider large squares. We first find a set which consists of polynomial number of subsets of large squares from L_{large}, and the optimal set $L_{large} \cap L^{opt}$ also belongs to it. Then, we show that there is a nice way of packing large squares inside an augmented unit square.

Selecting large squares. We say that a subset of large squares is *feasible* if it can be packed into the unit square. We can prove the following result.

Lemma 3.1. *In $O(n^{O(1)})$ time we can find the set $FEASIBLE$ consisting of all subsets of at most $1/\Delta^2$ large squares from L_{large}. Any feasible set of large squares belongs to $FEASIBLE$. Moreover, the optimal set of large squares $L_{large} \cap L^{opt}$ is a feasible set of large squares and, hence, it also belongs to $FEASIBLE$.*

Proof. Recall that the side of any large square is at least Δ. By definition, any feasible set of large squares can be packed into the unit square, i.e. into the area of size 1. The area of any large square is at least Δ^2. Hence, there are at most $1/\Delta^2$ large rectangles in any feasible set. There are at most n squares in L_{large}. So, we simply enumerate all

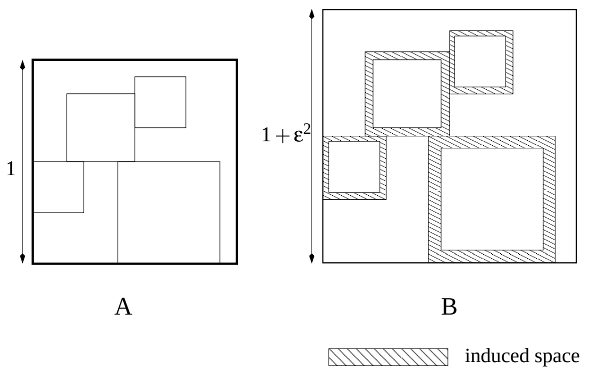

Figure 3: Increasing and decreasing the sizes of the large squares

subsets of $1/\Delta^2$ squares from L_{large}. This gives at most $O(n^{1/\Delta^2})$ sets. Notice that the optimal list $L_{large} \cap L^{opt}$ of large squares is also feasible. Hence, it must be among them. The lemma follows. □

Packing large squares. Even if we can find the optimal set of large squares, we still need to determine how to place these large squares in the unit square. We enlarge the size of the unit square so that there is a packing for the large squares, such that the positions of their left lower corners belong to a finite set of discrete points.

Lemma 3.2. *By augmenting the unit square to* $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$, *we can position every large square inside this augmented square such that its left lower corner is in the set*

$$CORNER = \{(x,y) | x = \ell \cdot (\varepsilon^3\Delta), y = p \cdot (\varepsilon^3\Delta) \text{ and } \ell, p = 1, 2, \ldots, 1/(\varepsilon^4\Delta)\}.$$

Proof. Take a packing in the unit square $[0,1] \times [0,1]$. In this packing, increase the size of each square by $(1+\varepsilon^2)$. Then, without reducing the size of the square, reduce every large square back to its original size. See Fig. 3 for an illustration of this process.

The side length of any large square is at least Δ. So, for each large square we now have an "induced space" where we can move the square up to a distance $\varepsilon^2\Delta$ vertically or horizontally, without increasing the area of the packing. Since $\varepsilon^2\Delta > \varepsilon^3\Delta$, we can move all large squares such that each one of them has its left lower corner in *CORNER*. □

4 Small Squares

Here we consider small squares. First, we need to select a set of small squares to pack, and then we show how to place them in an augmented unit square.

Selecting the small squares. Let $L'_{large} \subseteq L_{large}$ be any feasible set of large squares. The *complement* of L'_{large} is defined as the set $COM(L'_{large})$ of small squares which is selected by a FPTAS for the knapsack problem with accuracy $\varepsilon' := \varepsilon^2$, knapsack capacity $S := 1 - size(L'_{large})$, and set of items $LIST := L_{small}$. We can prove the following simple result.

Lemma 4.1. *For the optimal set* $L^{opt} \cap L_{large}$ *of large squares, its complement* $COM(L^{opt} \cap L_{large})$ *has total area at most*

$$1 - size(L^{opt} \cap L_{large})$$

and profit at least

$$(1-\varepsilon^2) profit(L^{opt} \cap L_{small}).$$

Proof. Notice that L^{opt} can be partitioned into the small squares in $L^{opt} \cap L_{small}$ and the large squares in $L^{opt} \cap L_{large}$ (we do not count the group $L^{(k)}$). The area of L^{opt} is at most 1. Hence, $L^{opt} \cap L_{small}$ is a feasible solution for the knapsack instance given by $S := 1 - size(L^{opt} \cap L_{large})$ and $LIST := L_{small}$. So, the optimum profit of this instance is at least the profit of $L^{opt} \cap L_{small}$. ¿From (1), the solution output by the FPTAS must have size at most S, and profit at least $(1-\varepsilon^2) profit(L^{opt} \cap L_{small})$. The lemma follows. □

Placing small squares: the modified NFDH. Assume that we have some feasible set $L'_{large} \subseteq L_{large}$ of large squares which is placed in the augmented unit square $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$. By solving a knapsack problem, we can find its complement $COM(L'_{large})$. Our next task is to place the small squares from $COM(L'_{large})$ in the augmented square $[0, 1+\varepsilon] \times [0, 1+\varepsilon]$.

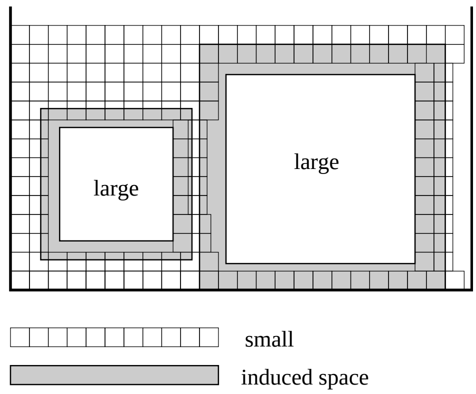

Figure 4: Packing the small squares

We use the modified NFDH heuristic: Recall that by using augmentation we have scaled up the unit square to $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$. The large squares from L'_{large} are placed inside it. We fill this augmented square with the small squares from $COM(L'_{large})$. As in the NDFH heuristic, we pack small squares on sublevels, bottom up and from left to right. On each sublevel, if the next small square overlaps with a large square, we simply place it immediately after the right boundary of the large square. For an illustration see Fig. 4. We cannot pack small squares within the space occupied by the large squares, but we can pack them inside the "induced space" around the large squares. We can prove the following result.

Lemma 4.2. *For any feasible set* $L'_{large} \subseteq L_{large}$ *of large squares packed in the augmented square* $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$, *the modified NFDH heuristic outputs a packing of* L'_{large} *and the small squares from its complement* $COM(L'_{large})$ *in the augmented square* $[0, 1+\varepsilon] \times [0, 1+\varepsilon]$.

Proof. Since we use the modified NFDH heuristic in each sublevel, at most one small square can cross the right border of the square $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$. Any small square has side at most $\varepsilon^4\Delta < \varepsilon^2$, hence, the total width of the packing is at most $(1+\varepsilon^2) + \varepsilon^2 < 1 + \varepsilon$.

Now we show that the height of the packing cannot be larger than $1+\varepsilon$. We follow the ideas of Lemma 2.1. Let H be the height of the packing. Let h_i $(i = 1, \ldots, q)$ be the height of the first square on the ith sublevel. Assume that

H is larger than $1+\varepsilon$. Consider one large square of side length s_i. Consider all sublevels ℓ that intersect it. Since we use the modified NFDH, the maximum distance from the large square's boundary to the closest small square on a sublevel ℓ cannot be larger than $\varepsilon^4\Delta$ (otherwise, a small square could be added on that sublevel). Hence, the maximum area not uncovered by small squares around, and including, this large square, is at most

$$(s_i + 2\varepsilon^4\Delta)^2.$$

Summing, over all large squares, we get that the area not covered by small squares is at most

$$\sum_{s_i \in L'_{large}} (s_i + 2\varepsilon^4\Delta)^2.$$

Notice that our packing for small squares goes further than point $1+\varepsilon^2$ in width, and $H = \sum_{i=1}^{q} h_i$. Then, as in Lemma 2.1, the area covered by all the squares from $COM(L'_{large})$ is

$$\begin{aligned}
AREA &\geq \sum_{i=1}^{q-1} h_{i+1}\cdot(1+\varepsilon^2) - \sum_{s_i\in L'_{large}} (s_i+2\varepsilon^4\Delta)^2 \\
&= (H-h_1)\cdot(1+\varepsilon^2) - \sum_{s_i\in L'_{large}} (s_i+2\varepsilon^4\Delta)^2 \\
&> (1+\varepsilon^2)^2 - \sum_{s_i\in L'_{large}} [(s_i^2 + 4s_i\varepsilon^4\Delta + (2\Delta\varepsilon^4)^2)] \\
&\quad \text{since } H > 1+\varepsilon \text{ and } h_1 < \varepsilon^4 \\
&\geq [1 - \sum_{s_i\in L'_{large}} s_i^2] + 2\varepsilon^2[1 - 2\varepsilon^2\Delta \sum_{s_i\in L'_{large}} s_i] + \\
&\quad + \varepsilon^4[1 - 4\Delta^2\varepsilon^4 |L'_{large}|].
\end{aligned} \tag{2}$$

Since $s_i \geq \Delta$ and $\varepsilon < 1/4$, then

$$1 - 2\varepsilon^2\Delta \sum_{s_i\in L'_{large}} s_i > 1 - \sum_{s_i\in L'_{large}} s_i^2 \geq 0.$$

From $|L'_{large}| \leq 1/\Delta^2$ we also get

$$1 - 4\Delta^2\varepsilon^4|L'_{large}| \geq 1 - 4\varepsilon^4 \geq 0.$$

Combining the above inequalities, we get

$$AREA > 1 - \sum_{s_i\in L'_{large}} s_i^2 = size(COM(L'_{large})).$$

This gives a contradiction. Hence, the value of H is at most $1+\varepsilon$.

The width of each sublevel is at most $1+\varepsilon$, and the height of the packing is at most $1+\varepsilon$. Hence, we have a packing in the augmented square $[0,1+\varepsilon]\times[0,1+\varepsilon]$, and the lemma follows. □

5 Overall Algorithm

Here we describe the complete algorithm.
ALGORITHM A_ε:
Input: A set of squares L, accuracy $\varepsilon > 0$.
Output: A packing of a subset of L in $[0,1+\varepsilon]\times[0,1+\varepsilon]$.

1. For each $k \in \{0,1\dots,1/\varepsilon^2\}$, form the group $L^{(k)}$ of squares with side lengths in $(\varepsilon^{4^{k+1}}, \varepsilon^{4^k}]$.
   (a) Define $\Delta := \varepsilon^{4^k}$.
   (b) Split $L \setminus L^{(k)}$ into L_{large} and L_{small}, the sets of large and small squares with side lengths larger than Δ and at most $\varepsilon^4\Delta$, respectively.
   (c) Compute the set $FEASIBLE$ containing all subsets of L_{large} with at most $1/\Delta^2$ large squares.
   (d) For every set $L'_{large} \in FEASIBLE$ find its complement $L'_{small} := COM(L'_{large})$ of small squares by solving a knapsack problem. For each packing of L'_{large} in the augmented square $[0,1+\varepsilon^2]\times[0,1+\varepsilon^2]$, such that every large square in L'_{large} has its left lower corner in a coordinate of $CORNER$:
      - Use the modified NFDH to pack the small squares L'_{small} in the augmented unit square $[0,1+\varepsilon]\times[0,1+\varepsilon]$.
2. Among all packings produced, select one with the largest profit, and output it.

Proof of Theorem 1.1: Algorithm A_ε only produces packings in the augmented square $[0,1+\varepsilon]\times[0,1+\varepsilon]$. Hence, we only need to compute the profit of the packing chosen in step 2. As we know, the optimal set of large squares $L^{opt}\cap L_{large}$ belongs to $FEASIBLE$. Furthermore, there exists a packing of these squares in the augmented square $[0,1+\varepsilon^2]\times[0,1+\varepsilon^2]$ such that each large square has its left lower corner in a coordinate of $CORNER$.

Since the algorithm A_ε checks all such possible packings, it will find one for $L^{opt}\cap L_{large}$. Next, A_ε finds the complement $COM(L^{opt}\cap L_{large})$ and packs it using the modified NFDH. The profit of the packing output by the algorithm is

$$\begin{aligned}
A_\varepsilon(L) &\geq profit(L^{opt}\cap L_{large}) + \\
&\quad profit(COM(L^{opt}\cap L_{large})) \\
&\geq profit(L^{opt}\cap L_{large}) + \\
&\quad (1-\varepsilon^2)profit(L^{opt}\cap L_{small}), \text{ by Lemma 4.1} \\
&\geq (1-\varepsilon^2)profit(L^{opt}\cap[L_{large}\cup L_{small}]) \\
&\geq (1-\varepsilon^2)[(1-\varepsilon^2)profit(L^{opt})], \text{ by Corollary 2.4} \\
&\geq (1-\varepsilon)\mathrm{OPT}.
\end{aligned}$$

We know that any set of large squares from $FEASIBLE$ consists of at most $(1/\Delta^2)$ squares. Hence, $FEASIBLE$ can be computed in (n^{1/Δ^2}) time, and we need to do this $1/\varepsilon^2$ times (once for each value of k, see Step 1 of the algorithm). Since $|CORNER| = \frac{1}{\varepsilon^8\Delta^2}$, at most $\left(\frac{1}{\varepsilon^8\Delta^2}\right)^{1/\Delta^2}$ time is required to produce all possible packings of large squares in the augmented square $[0,1+\varepsilon^2]\times[0,1+\varepsilon^2]$. The running time of the FPTAS in (Lawler 1979) for the knapsack problem is $O(n^2\log(n/\varepsilon))$. The modified NFDH algorithm runs in $O(n\log n)$ time. Combining all together, we get that the running time of the algorithm is

$$O\left(\frac{n^2}{\varepsilon^2}\log(n/\varepsilon)\left(\frac{n}{\varepsilon^8\Delta^2}\right)^{1/\Delta^2}\right),$$

where $\Delta = \varepsilon^{4^{1/\varepsilon^2}}$. The theorem follows.

Conclusions

In this paper we have presented an algorithm for packing a set $L' \subseteq L$ of squares into a square region of size $1+\varepsilon$, for any given $\varepsilon > 0$. The profit of L' is at least $(1-\varepsilon)\mathrm{OPT}(L)$, where $\mathrm{OPT}(L)$ is the maximum profit that can be achieved by packing squares from L into a unit size square region. Although this problem is related to the 2-dimensional bin packing problem (Bansal & Sviridenko 2004, Correa & Kenyon 2004), in which a set of rectangles must be packed into the smallest possible number of unit square bins, the goals of the two problems are sufficiently different that the algorithms described in (Bansal & Sviridenko 2004, Correa & Kenyon 2004) do not extend to our problem, and so, new ideas are required.

An interesting open problem is that of finding a set $L' \subseteq L$ of squares with profit at least $(1-\varepsilon)\mathrm{OPT}(L)$ and a packing for them in the unit square region $[0,1]\times[0,1]$. Natural extensions of our algorithm (like removing one of the large squares to accommodate those squares that in our algorithm would overflow the boundaries of the unit square region, thus, requiring the ε extension in the size of the region) do not work. We conjecture that this this more complex problem can be solved, but new techniques seem to be needed.

References

Baker, B., Brown, D. & Katseff, H. (1981), 'A 5/4 algorithm for two dimensional packing', *Journal of Algorithms* **2**, 348–368.

Baker, B., Calderbank, A., Coffman, E. & Lagarias, J. (1983), 'Approximation algorithms for maximizing the number of squares packed into a rectangle', *SIAM Journal on Algebraic and Discrete Methods* **4**, 383–397.

Baker, B., Coffman, E. & Rivest, R. (1980), 'Orthogonal packings in two dimensions', *SIAM Journal on Computing* **9**, 846–855.

Bansal, N. & Sviridenko, M. (2004), New approximability and inapproximability results for 2-dimensional bin packing, *in* 'Proceedings 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)', pp. 189–196.

Caprara, A. (2002), Packing 2-dimensional bins in harmony, *in* 'Proceedings 43rd Annual Symposium on Foundations of Computer Science(FOCS)', pp. 490–499.

Chung, F., Garey, M. & Johnson, D. (1982), 'On packing two-dimentional bins', *SIAM Journal on Algebraic and Discrete Methods* **3**, 66–76.

Coffman, E., Garey, M., Johnson, D. & Tarjan, R. (1980), 'Performance bounds for level-oriented two-dimensional packing algorithms', *SIAM Journal on Computing* **9**, 808–826.

Correa, J. & Kenyon, C. (2004), Approximation schemes for multidimensional packing, *in* 'Proceedings 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)', pp. 179–188.

Ferreira, C., Miyazawa, F. & Wakabayashi, Y. (1999), 'Packing squares into squares', *Perquisa Operacional* **19**, 349–355.

Garey, M. R. & Johnson, D. S. (1979), *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, San Francisco, CA.

Gilmore, P. & Gomory, R. (1965), 'Multistage cutting stock problems of two and more dimensions', *Operations Research* **13**, 94–120.

Golan, I. (1981), 'Performance bounds for orthogonal, oriented two-dimensional packing algorithms', *SIAM Journal on Computing* **10**, 571–582.

Jansen, K. & Zhang, G. (2004*a*), Maximizing the number of packed rectangles, *in* 'Proceedings 9th Scandinavian Workshop on Algorithm Theory (SWAT)'. To appear.

Jansen, K. & Zhang, G. (2004*b*), On rectangle packing: maximizing benefits, *in* 'Proceedings 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)', pp. 197–206.

Kellerer, H., Pferschy, U. & Pisinger, D. (2004), *Knapsack problems*, Springer.

Kenyon, C. & Rémila, E. (1996), Approximate strip-packing, *in* 'Proceedings 37th Annual Symposium on Foundations of Computer Science (FOCS)', pp. 31–36.

Kleitman, D. & Krieger, M. (1975), An optimal bound for two dimensional bin packing, *in* 'Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science (FOCS)', pp. 163–168.

Lawler, E. (1979), 'Fast approximation algorithms for knapsack problems', *Mathematics of Operations Research* **4**, 339–356.

Leung, J.-T., Tam, T., Wong, C., Young, G. & Chin, F. (1990), 'Packing squares into a square', *Journal of Parallel and Distributed Computing* **10**, 271–275.

Meir, A. & Moser, L. (1968), 'On packing of squares and cubes', *Journal of Combinatorial Theory* **5**, 126–134.

Moon, J. & Moser, L. (1967), 'Some packing and covering theorems', *Colloquium Mathematicum* **17**, 103–110.

Moser, L. (1965), Poorly formulated unsolved problems of combinatorial geometry. Mimeographed.

Moser, W. & Pach, J. (1989), Problem N.108, *in* 'Research Problems in Discrete Geometry', McGill University, Montreal.

Novotny, P. (1995), A note on packing of squares, *in* 'Studies of University of Transport and Communications in Zilina', number 10 *in* 'Math.-Phys. series'.

Novotny, P. (1996), 'On packing of squares into a rectangle', *Archivum Mathematicum* **32**, 75–83.

Schiermeyer, I. (1994), Reverse fit : a 2-optimal algorithm for packing rectangles, *in* 'Proceedings 2nd European Symposium on Algorithms (ESA)', pp. 290–299.

Seiden, S. & van Stee, R. (2003), 'New bounds for multidimentional packing', *Algorithmica* **36(3)**, 261–293.

Sleator, D. (1980), 'A 2.5 times optimal algorithm for bin packing in two dimensions', *Information Processing Letters* **10**, 37–40.

Steinberg, A. (1997), 'A strip-packing algorithm with absolute performance bound 2', *SIAM Journal on Computing* **26(2)**, 401–409.