Algorithmic Operations Research Vol.3 (2008) 1–12

On Packing Rectangles with Resource Augmentation: Maximizing the Profit

Aleksei V. Fishkin

Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany.

Olga Gerber

University of Kiel, Olshausenstr. 40, 24118 Kiel, Germany.

Klaus Jansen

University of Kiel, Olshausenstr. 40, 24118 Kiel, Germany.

Roberto Solis-Oba

Department of Computer Science, The University of Western Ontario, London, Ontario, N6A 5B7, Canada.

Abstract

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

Key words: Rectangle packing, approximation algorithms, resource augmentation

1. Introduction

There has recently been an increasing interest in solving a variety of 2-dimensional packing problems such as strip packing [18,28,32], 2-dimensional bin packing [4–6,29], and rectangle packing [1,2,16]. 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 rectangles with profits into a unit size square region so as to maximize the total profit of the packed rectangles. More precisely, we are given a set R of n rectangles, R_i ($i = 1,\dots,n$) with widths $a_i \in (0,1]$, heights $b_i \in (0,1]$, and profits $p_i \geq 0$. For a given subset $R' \subseteq R$, a *packing* of R' into a unit size square frame $[0,1]\times[0,1]$ is a positioning of the rectangles of R' within the frame such that they have disjoint interiors. The goal is to find a subset $R' \subseteq R$, and a packing of R' within $[0,1]\times[0,1]$ of maximum profit, $\sum_{R_i\in R'} p_i$. We only consider the version of the problem when rotations of the rectangles are not allowed. Therefore, by scaling the sizes of the rectangles, it is easy to show that the above problem is equivalent to the problem of packing a set R of rectangles into a rectangular frame of width $a > 0$ and height $b > 0$.

This problem is known to be strongly NP-hard even for the restricted case of packing squares with identical profits [21]. Hence, it is very unlikely that any polynomial time algorithm for the 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 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) for a maxi-

Email: Aleksei V. Fishkin [avf@mpi-sb.mpg.de], Olga Gerber [oge@informatik.uni-kiel.de], Klaus Jansen [kj@informatik.uni-kiel.de], Roberto Solis-Oba [solis@csd.uwo.ca].

mization problem Π is a family of approximation algorithms $\{A_\varepsilon\}_{\varepsilon>0}$ such that A_ε is a $(1-\varepsilon)$-approximation algorithm for Π 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 rectangle packing problem is equivalent to the knapsack problem: given a knapsack of capacity B and a set of items with profits and sizes, pack a subset of 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 [11], and it admits a FPTAS [17,20]. In contrast, our problem is strongly NP-hard, and, hence, it admits no FPTAS unless P = NP.

For the 2-dimensional version of the problem, one can see a relationship to the problem of packing squares into a rectangle of minimum area [24,25]: Find the minimum value x such that any set of squares of total area 1 can be packed into a rectangle of area x. Regarding lower bounds for this latter problem, there is just one non-trivial result known [26]: 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 algorithms yielding several upper bounds for this problem. As it was shown in [23], 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 [22], 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. Further results in this direction were obtained in [19], where 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 [27] shows that any set L of squares of total area 1 can be packed into a rectangle whose area is less than 1.53.

Our problem is also related to the 2-dimensional bin packing problem: Given a set L of rectangles of specified size (width and height), pack them into the minimum number of unit size square bins. The problem is strongly NP-hard [21] and no approximation algorithm for it has approximation ratio smaller than 2, unless P = NP [10]. A long history of approximation results exists for this problem and its variants [4–6,29]. Very recently a number of asymptotic results have been obtained for it (i.e. for the case when the optimum uses a large number of bins). In [4] it was proven that the general version of the problem does not admit an asymptotic PTAS, unless P = NP. However, there is an asymptotic PTAS if all rectangles are actually squares [4,8]. Also, in [8] a polynomial algorithm was presented which packs any set L of rectangles into at most $N^{opt}(L)$ augmented bins of size $(1+\varepsilon)$ for any $\varepsilon > 0$, where $N^{opt}(L)$ denotes the minimum number of unit size bins required to pack the rectangles in L.

A related problem is the two-dimensional knapsack problem [9] in which a set of rectangular pieces needs to be cut off a rectangular plate of width a and height b. Each rectangular piece R_i has width a_i, height b_i, and profit p_i, and an arbitrary number of pieces of type R_i can be cut from the plate. The goal is to cut the plate so as to maximize the total profit of the pieces produced.

Finally, one can also see a relationship to strip packing [12]: Given a set L of rectangles, it is required to pack them 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 strip packing ideas come from bin packing. The "Bottom-Left" heuristic has asymptotic performance ratio 2 when the rectangles are sorted by decreasing widths [3]. In [7] several simple algorithms were studied that place the rectangles 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. The asymptotic performance ratio was further reduced to $3/2$ [31], then to $4/3$ [13], and to $5/4$ [1]. Finally, in [18] 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 performance ratio 2 [28,32].

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 [2]. Only very recently this algorithm for packing unit profit squares was improved to a PTAS [15]. For packing rectangles with profits, several approximation algorithms were presented in [16]. The best one is a $(\frac{1}{2} - \varepsilon)$-approximation algorithm, for any fixed $\varepsilon > 0$.

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

Theorem 1. *For any set R of n rectangles and any accuracy $\varepsilon > 0$, there is an algorithm W_ε which finds a subset of R and its packing within an augmented unit square frame,* $[0,1+3\varepsilon] \times [0,1+3\varepsilon]$*, with profit*

$$W_\varepsilon(R) \geq (1-\varepsilon)\mathrm{OPT},$$

where OPT *is the maximum profit that can be obtained by packing any subset of R into a unit size square frame* $[0,1] \times [0,1]$*. The running time of W_ε is polynomial in n for fixed ε.*

We note that the algorithm of Correa and Kenyon [8] for packing a set of rectangles into the minimum number of square bins of size $1+\varepsilon$ can not be directly used to prove Theorem 1 because (i) the algorithm in [8] does not consider rectangles with profits, and (ii) in the rectangle packing problem not all rectangles need to be packed. If we can find a set of rectangles of nearly maximum profit and which can be packed into a unit square frame, then we could use the algorithm in [8] to find such a packing. The problem of finding this set of rectangles is not a simple one, though. We show how to find in polynomial time a set of rectangles of nearly optimum profit that can be packed into a square frame of size $1+\varepsilon$. This is enough to prove the theorem.

We first address the special case of the problem when all rectangles to be packed are squares. Presenting the algorithm for this simpler problem will help to understand the solution for the more complex problem of packing rectangles. Specifically, we present an algorithm A_ε which given a set of squares L finds a subset of L and its packing into the augmented unit square $[0,1+\varepsilon] \times [0,1+\varepsilon]$ with profit

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

where OPT 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 polynomial in n for fixed ε. This result can be extended to the case of packing d-dimensional cubes into a d-dimensional cube of size $1+\varepsilon$, for $d \geq 2$.

Our algorithms combine several known approximation techniques used for knapsack problems, strip packing, and scheduling problems. Our algorithm for packing squares is based on a few simple ideas and, contrasting to recent algorithms for packing problems [4,8,16,18], it does not use linear programming. Since the problem for packing squares is a special case of that of packing rectangles, our algorithm is simpler and more efficient that the algorithm in [8]. The algorithm deals separately with squares of different sizes. This idea has been used before to solve other problems [14,30]. We partition the squares into two sets formed by large and small squares, respectively. The sets are chosen so that only $O(1)$ large squares can be packed in the unit square frame. We augment the size of the frame to $1+\varepsilon$, and discretize the set of possible positions for the large squares in a packing. This allows us to enumerate all possible packings of the large squares. For each one of these packing we try to fill with small squares the empty spaces left by the large squares. To do this we solve a knapsack problem to select the small squares to be packed, and use a variation of the Next-Fit-Decreasing-Height heuristic to place them (see Section 2.1.). Among all packings found we select one with the maximum profit, which must be at least $(1-\varepsilon)\mathrm{OPT}$.

For the problem of packing rectangles we need to make a more complex partition, separating the rectangles into four groups: $\mathcal{L}$,$\mathcal{H}$,$\mathcal{V}$, and $\mathcal{S}$. Sets $\mathcal{L}$ and $\mathcal{S}$ contain rectangles with, respectively, large and small widths and heights. These are treated in a similar way as above. The other two sets, $\mathcal{H}$ and $\mathcal{V}$, contain wide and short (i.e. horizontal), and narrow and tall (i.e. vertical) rectangles, respectively. To pack these rectangles we first round their sizes and group them, so they form larger rectangles. These grouped rectangles are then packed by solving a fractional strip packing problem.

Even though the running times of both algorithms A_ε and W_ε are polynomial in n for fixed ε, they are exponential in $1/\varepsilon$. Therefore, our results are primarily of theoretical importance.

In Section 2. we describe our algorithm for packing squares. In Section 3. we describe an algorithm for packing a set of rectangles into an augmented square frame and we give a proof for Theorem 1. Finally, in the last section we give some concluding remarks.

2. Algorithm for Packing Squares

In this section we present an algorithm for packing squares into a unit size square frame so as to maximize the total profit of the packed squares. More precisely, we are given a set Q 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 $Q' \subseteq Q$, a *packing* of Q' into the unit square is a positioning of the squares Q' within the frame $[0,1] \times [0,1]$ such that they have disjoint interiors. The goal is to find a subset $Q' \subseteq Q$ and its packing into the unit square, of maximum profit, $\sum_{S_i \in Q'} p_i$.

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

$$profit(Q^{opt}) = \text{OPT and } area(Q^{opt}) \leq 1.$$

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

2.1. *The NFDH Heuristic*

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

To solve this problem, we sort the squares of Q' non-increasingly 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 at the top of the largest square placed on the previous sublevel. In each sublevel, squares are packed in a left-justified manner until their total width is at least a. At that moment, the current sublevel is closed, a new sublevel is started and the packing proceeds as above. For an illustration see Fig. 1.

We will use the following simple result, which can be directly derived from results in [7,22], but for completeness we include a proof.

Lemma 2. *Let $Q' \subseteq Q$ be any subset of squares with side lengths at most ε^2, ordered non-increasingly by side lengths, and let $[0,a] \times [0,b]$ $(a, b \in [0,1])$ be a rectangle such that $area(Q') \leq ab$. Then, the NFDH heuristic outputs a packing of Q' 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. Since

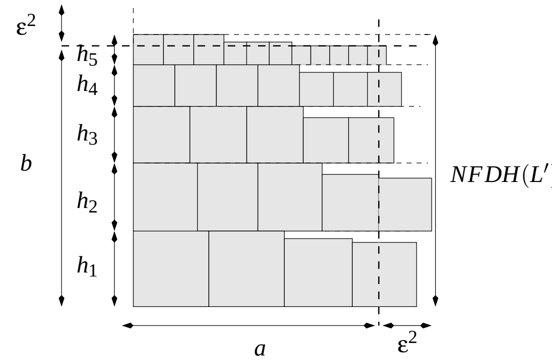

Fig. 1. NFDH for small squares.

NFDH packs the squares of Q' on sublevels in order of non-increasing side lengths, 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 sublevel (except, maybe, the last) is at least a and at most $a + \varepsilon^2$. Then, the total area of the squares on the ith sublevel $(i = 1, \ldots, q-1)$ 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

$$\sum_{i=1}^{q-1} h_{i+1} \cdot a = a \cdot \sum_{i=2}^{q} h_i$$
$$= a[H - h_1] > a[(b+\varepsilon^2) - h_1] \text{ by assumption } H > b+\varepsilon^2$$
$$= a[b + (\varepsilon^2 - h_1)] \geq ab = area(Q') \text{ since } h_1 \leq \varepsilon^2,$$

which gives a contradiction.

□

Collorary 3. *If all squares in Q have side length at most ε^2, then there is an algorithm which finds a subset of Q 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 Q, whose total area is at most 1 and whose profit is at least $(1-\varepsilon)$OPT. By using NFDH we can pack these squares into the augmented frame $[0, 1+\varepsilon^2] \times [0, 1+\varepsilon^2]$. □

2.2. *Partitioning the Squares*

Now we consider the case of squares with arbitrary sizes. 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,

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

We will use the following simple observation, which also has been made by other researchers in different contexts [4,8,14,30].

Lemma 4. *There is a group $L^{(k)}$ with $0 \leq k \leq 1/\varepsilon^2 - 1$ such that its contribution to the optimum is*

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

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

Proof. Since $L^{(\ell)} \cap L^{(j)} = \emptyset$ for all $\ell \neq j$, then

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

There must exist at least one group $L^{(k)}$ with $0 \leq k \leq 1/\varepsilon^2 - 1$ whose contribution to the profit of the optimal solution is at most the average contribution of the $1/\varepsilon^2$ groups:

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

□

We drop the squares in this group $L^{(k)}$ of low profit from consideration. Then, an optimal packing for $Q \setminus L^{(k)}$ 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 partition the squares in $Q \setminus L^{(k)}$ into two groups: $\mathcal{L} = \cup_{j \leq k-1} L^{(j)}$ and $\mathcal{S} = \cup_{j \geq k+1} L^{(j)}$. The squares in $\mathcal{L}$ and $\mathcal{S}$ are called large and small, respectively.

Collorary 5. *Let $\Delta = \varepsilon^{4^k}$, where k is as defined above. The side length of any large square is larger than Δ and the side length of any small square is at most $\varepsilon^4\Delta$. Moreover,*

$$profit(Q^{opt} \cap [\mathcal{L} \cup \mathcal{S}]) \geq (1-\varepsilon^2)\text{OPT}.$$

2.3. *Large Squares*

We say that a subset of large squares is *feasible* if it can be packed into the unit square frame. 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. Let $FEASIBLE$ be the set consisting of all subsets of at most $1/\Delta^2$ large squares from $\mathcal{L}$. Since there are at most *n* squares in $\mathcal{L}$, there is only a polynomial number, $O(n^{1/\Delta^2})$, of sets in $FEASIBLE$. Note that $\mathcal{L} \cap Q^{opt}$ must belong to $FEASIBLE$.

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

Consider a packing of a subset of large squares in the frame $[0,1] \times [0,1]$. In this packing, increase the size of each large square by a factor $1+\varepsilon^2$. This increases the size of the enclosing frame by the same factor. Then, without reducing the size of the frame, reduce the size of every large square back to its original value. See Fig. 2 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 lower left corner in the following set

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

By discretizing the positions of the large squares we reduce to a constant the number of different packings for the large squares in a feasible set.

2.4. *Small Squares*

Let $\mathcal{L}' \subseteq \mathcal{L}$ be any feasible set of large squares. The *complement* of $\mathcal{L}'$, denoted $COM(\mathcal{L}')$, is the set of small squares which is selected by a FPTAS [17] for the knapsack problem with accuracy ε^2, knapsack capacity $1 - area(\mathcal{L}')$, and set of items $\mathcal{S}$; each item $S_i \in \mathcal{S}$ has size $(s_i)^2$ and profit p_i. We can prove the following simple result.

Lemma 6. *For the optimal set $Q^{opt} \cap \mathcal{L}$ of large squares, its complement $COM(Q^{opt} \cap \mathcal{L})$ has total area at most*

$$1 - area(L^{opt} \cap \mathcal{L})$$

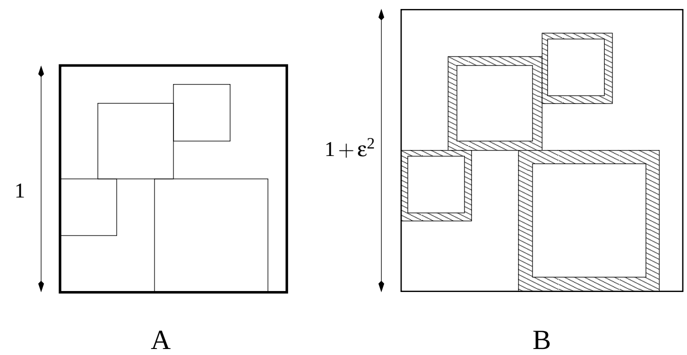

Fig. 2. Increasing and decreasing the sizes of the large squares.

and profit at least

$$(1-\varepsilon^2)profit(Q^{opt}\cap\mathcal{S}).$$

Proof. The area of Q^{opt} is at most 1, hence, $Q^{opt}\cap\mathcal{S}$ is a feasible solution for the instance of the knapsack problem with knapsack capacity $1-area(Q^{opt}\cap\mathcal{L})$ and set of items $\mathcal{S}$. So, the optimum profit of this instance is at least $profit(Q^{opt}\cap\mathcal{S})$ and the FPTAS finds a solution of profit at least $(1-\varepsilon^2)profit(Q^{opt}\cap\mathcal{S})$. □

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

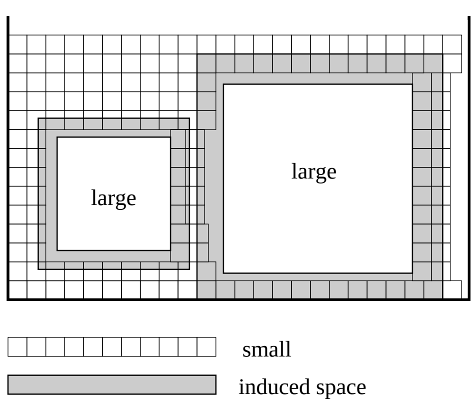

Fig. 3. Packing the small squares.

We pack the small squares in the empty space left by the large squares using the modified NFDH heuristic from [7]: Pack the squares on sublevels, creating sublevels in a bottom up manner and filling each one of them from left to right. On each sublevel, if the next small square overlaps with a large square, we place it immediately after the right boundary of the large square. For an illustration see Fig. 3. 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 7. *For any feasible set $\mathcal{L}'\subseteq\mathcal{L}$ of large squares packed in the augmented frame $[0,1+\varepsilon^2]\times[0,1+\varepsilon^2]$, the modified NFDH heuristic outputs a packing of $\mathcal{L}'$ and the small squares from its complement $COM(\mathcal{L}')$ in the augmented frame $[0,1+\varepsilon]\times[0,1+\varepsilon]$.*

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 $(1+\varepsilon^2)+\varepsilon^2<1+\varepsilon$, for $\varepsilon<1/4$.

Now we show that the height of the packing cannot be larger than $1+\varepsilon$. We follow the ideas of Lemma 2. 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. We assume that H is larger than $1+\varepsilon$ and derive a contradiction. Consider one large square of side length s_i and all sublevels ℓ that intersect it. 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 covered 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

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

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, the area covered by the squares from $COM(\mathcal{L}')$ is

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

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

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

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

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

Combining the above inequalities, we get

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

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

2.5. *The Algorithm*

ALGORITHM A_ε:
Input: A set of squares Q, accuracy $\varepsilon > 0$.
Output: A packing of a subset of Q in $[0,1+\varepsilon] \times [0,1+\varepsilon]$.

(1) For each $k \in \{0,1\ldots,1/\varepsilon^2\}$, form the group $L^{(k)}$ as described above.
   (a) Let $\Delta := \varepsilon^{4^k}$.
   (b) Split $Q \setminus L^{(k)}$ into $\mathcal{L}$ and $\mathcal{S}$, 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 $\mathcal{L}$ with at most $1/\Delta^2$ large squares.
   (d) For every set $\mathcal{L}' \in FEASIBLE$ find its complement $\mathcal{S}' := COM(\mathcal{L}')$ by solving a knapsack problem. For each packing of $\mathcal{L}'$ in the augmented square $[0,1+\varepsilon^2] \times [0,1+\varepsilon^2]$ such that every large square in $\mathcal{L}'$ has its lower left corner in a point of $CORNER$:
      - Use the modified NFDH to pack the small squares $\mathcal{S}'$ 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.

Theorem 8. *For any set Q of n squares and any fixed value $\varepsilon > 0$, there exists an algorithm A_ε which finds a subset of Q and its packing into the augmented unit square $[0,1+\varepsilon] \times [0,1+\varepsilon]$ with profit*

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

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

$$O\left(\frac{n^2}{\varepsilon^3}\left(\frac{n}{\varepsilon^8\Delta^2}\right)^{1/\Delta^2}\right),$$

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

Proof. By Lemma 7 algorithm A_ε produces a packing 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. The optimal set of large squares $Q^{opt} \cap \mathcal{L}$ belongs to $FEASIBLE$, and hence, 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 lower left corner in a point of $CORNER$.

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

$$\begin{aligned}
A_\varepsilon(Q) &\geq profit(Q^{opt} \cap \mathcal{L}) + profit(COM(Q^{opt} \cap \mathcal{L})) \\
&\geq profit(Q^{opt} \cap \mathcal{L}) + (1-\varepsilon^2) profit(Q^{opt} \cap \mathcal{S}) \\
&\text{(by Lemma 6)} \\
&\geq (1-\varepsilon^2) profit(Q^{opt} \cap [\mathcal{L} \cup \mathcal{S}]) \\
&\geq (1-\varepsilon^2)[(1-\varepsilon^2) profit(Q^{opt})] \\
&\text{(from Corollary 5)} \\
&\geq (1-\varepsilon)\mathrm{OPT}.
\end{aligned}$$

We know that any set of large squares from $FEASIBLE$ consists of at most $(1/\Delta^2)$ squares. Hence, $FEASIBLE$ can be computed in $O(n^{1/\Delta^2})$ time, and we need to do this $1/\varepsilon^2$ times (once for each value of k, see Step 1 of the algorithm). Since $|CORNER| = (\frac{1+\varepsilon^2-\Delta}{\varepsilon^3\Delta})^2 \leq \frac{1}{\varepsilon^8\Delta^2}$, the algorithm computes at most $\left(\frac{1}{\varepsilon^8\Delta^2}\right)^{1/\Delta^2}$ packings of large squares in the augmented square $[0,1+\varepsilon^2]\times[0,1+\varepsilon^2]$. The running time of the basic-FPTAS in [17] for the knapsack problem is $O(n^2\cdot 1/\varepsilon)$ (the different versions of FPTAS can be found in [17]). The modified NFDH algorithm runs in $O(n\log n)$ time. Combining all together, we get that the running time of the algorithm is

$$O\left(\frac{(n^{1/\Delta^2})}{\varepsilon^2}\cdot\left(\frac{1}{\varepsilon^8\Delta^2}\right)^{1/\Delta^2}\left[(n^2\cdot 1/\varepsilon))+(n\log n)\right]\right).$$

Simplifying, we find that the running time of the overall algorithm is bounded by

$$O\left(\frac{n^2}{\varepsilon^3}\left(\frac{n}{\varepsilon^8\Delta^2}\right)^{1/\Delta^2}\right),$$

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

2.6. *Packing d-Dimensional Cubes*

Our algorithm can be easily extended to the problem of packing d-dimensional cubes into a unit d-dimensional cubic frame so as to maximize the total profit of the cubes packed. As in the 2-dimensional case, we partition the set of cubes into two sets $\mathcal{L}$ and $\mathcal{S}$ containing large and small cubes, respectively. Since only a constant number of large cubes can be packed into the frame, we can enumerate all feasible subsets of $\mathcal{L}$ that can be packed in the augmented cubic frame of size $1+\varepsilon^2$ in polynomial time. We can prove the following generalization of Lemma 2 (see also [8]).

Lemma 9. *Let $Q' \subseteq Q$ be any subset of d-dimensional cubes with side lengths at most ε^2, ordered by non-increasing side lengths, and let $[0,a_1]\times[0,a_2]\times\cdots\times[0,a_d]$ ($a_i\in[0,1]$) be a parallelepiped, such that $area(Q')\leq a_1\times a_2\ldots\times a_d$. Then, the generalization of the NFDH heuristic to d dimensions outputs a packing of Q' in the augmented parallelepiped $[0,a_1+\varepsilon^2]\times[0,a_2+\varepsilon^2]\times\cdots\times[0,a_d+\varepsilon^2]$.*

This lemma shows that the generalization of NFDH to d dimensions can be used to pack the small cubes in the empty spaces left by a packing of the large cubes into the augmented cubic frame. Then, we can prove that the generalization of the modified NFDH heuristic to d dimensions outputs a packing of $\mathcal{L}'$ and the small cubes from its complement $COM(\mathcal{L}')$ in the augmented cubic frame of size $1+\varepsilon$. Among all packings found we select one with the maximum profit, which must be at least $(1-\varepsilon)$OPT.

3. Algorithm for Packing Rectangles

Let R be a set of n rectangles, R_i $(i=1,\ldots,n)$ with widths $a_i\in(0,1]$, heights $b_i\in(0,1]$, and profits $p_i\geq 0$. The goal is to find a subset $R'\subseteq R$, and a packing of R' within the frame $[0,1]\times[0,1]$ of maximum profit, $\sum_{R_i\in R'} p_i$.

We partition the rectangles R into four sets: $\mathcal{L}$,$\mathcal{H}$,$\mathcal{V}$, and $\mathcal{S}$. The rectangles in $\mathcal{L}$ have large widths and heights, so only $O(1)$ of them can be packed in the unit square frame. The rectangles in $\mathcal{H}$ ($\mathcal{V}$) have large width (height). We round the sizes of these rectangles in order to reduce the number of distinct widths and heights. Then, we use enumeration and a fractional strip-packing algorithm to select the best subsets of $\mathcal{H}$ and $\mathcal{V}$ to include in our solution. The rectangles in $\mathcal{S}$ have very small width and height, so as soon as we have selected near-optimal subsets of rectangles from $\mathcal{L}\cup\mathcal{H}\cup\mathcal{V}$ we add rectangles from $\mathcal{S}$ to the set of rectangles to be packed in a greedy way. Once we have selected the set of rectangles to be packed into the frame, we use a slight modification of the algorithm of Correa and Kenyon [8] to pack them.

For a subset of rectangles $R'\subseteq R$, we use $profit(R')$ to denote its profit, $\sum_{R_i\in R'} p_i$, and $area(R')$ to denote its area, $\sum_{R_i\in R'} a_i b_i$. In addition, we use R^{opt} to denote an optimal subset of R that can be packed into the unit square frame $[0,1]\times[0,1]$. So,

$$profit(R^{opt}) = \text{OPT and } area(R^{opt})\leq 1.$$

3.1. *Partitioning the Rectangles*

We slightly modify the definition of the groups $L^{(j)}$ given above to account for the fact that now the width and height of a rectangle might be different. We define the group $L^{(0)}$ of rectangles $R_i\in R$ with widths $a_i\in(\varepsilon^4,1]$ and/or heights $b_i\in(\varepsilon^4,1]$. For $j\in Z_+$ we define the group $L^{(j)}$ of rectangles R_i with either widths $a_i\in(\varepsilon^{4^{j+1}},\varepsilon^{4^j}]$ or heights $b_i\in(\varepsilon^{4^{j+1}},\varepsilon^{4^j}]$. One can see that each rectangle belongs to at most 2 groups.

Lemma 10. *There is a group $L^{(k)}$ with $0 \le k \le 2/\varepsilon^2 - 1$ such that*

$$profit(L^{(k)} \cap R^{opt}) \le \varepsilon^2 \cdot \mathrm{OPT},$$

where R^{opt} is the subset of rectangles selected by an optimum solution.

Proof. The proof is very similar to the proof of Lemma 4 □

We again drop the rectangles in group $L^{(k)}$, as described in Lemma 10, from consideration. Then, an optimal packing for $R^{opt} \setminus L^{(k)}$ must have profit at least $(1-\varepsilon^2)$OPT. However, now we partition the rectangles of R into four groups according to their side lengths, as follows. Let $\Delta = \varepsilon^{4^k}$.

$$\mathcal{L} = \{R_i \mid a_i > \Delta \text{ and } b_i > \Delta\}$$
$$\mathcal{S} = \{R_i \mid a_i \le \varepsilon^4\Delta \text{ and } b_i \le \varepsilon^4\Delta\}$$
$$\mathcal{H} = \{R_i \mid a_i > \Delta \text{ and } b_i \le \varepsilon^4\Delta\}$$
$$\mathcal{V} = \{R_i \mid a_i \le \varepsilon^4\Delta \text{ and } b_i > \Delta\}$$

Lemma 11. *For $0 < \varepsilon < 1/2$ the subset $R^{opt} \setminus L^{(k)}$ of rectangles can be packed within the frame $[0, 1+\varepsilon] \times [0, 1+\varepsilon]$ in such a way that*

- *each rectangle $R_i \in \mathcal{H} \cup \mathcal{L}$ is positioned so that its lower left corner is at an x-coordinate that is a multiple of $\varepsilon^2\Delta$,*
- *each rectangle $R_i \in \mathcal{V} \cup \mathcal{L}$ is positioned so that its lower left corner is at a y-coordinate that is a multiple of $\varepsilon^2\Delta$,*

Furthermore, any width $a_i > \Delta$ or height $b_i > \Delta$ can be rounded up to the nearest multiple of $\varepsilon^2\Delta$ without affecting the feasibility of the packing, i.e. (i) for each $R_i \in \mathcal{L}$, both, a_i and b_i can be rounded up, (ii) for each $R_i \in \mathcal{H}$, only a_i can be rounded, and (iii) for each $R_i \in \mathcal{V}$, only b_i can be rounded.

Proof. Increase the size of every rectangle in $\mathcal{L} \cup \mathcal{H} \cup \mathcal{V}$ by a factor $1+\varepsilon$. These enlarged rectangles can be packed in a frame of size $1+\varepsilon$. Now shrink the rectangles back to their original sizes to create the "induced spaces" as before. Shift each rectangle inside its induced space so that it is positioned as indicated in the lemma. Note that each rectangle needs to be shifted vertically and/or horizontally at most a distance $\varepsilon^2\Delta$. Finally, round each side length larger than Δ to the nearest multiple of $\varepsilon^2\Delta$. Since each rectangle can be shifted inside its induced space vertically or horizontally by a distance $\varepsilon\Delta$, and since $2\varepsilon^2\Delta < \varepsilon\Delta$ for all $0 < \varepsilon < 1/2$, then the enlarged rectangles fit in a frame of size $1+\varepsilon$. □

Selecting the large rectangles. As before, we say that a subset of large rectangles is feasible if they can be packed in the unit frame. We define the set $FEASIBLE$ consisting of all subsets of at most $1/\Delta^2$ large rectangles. Observe that the optimal set of large rectangles $\mathcal{L} \cap R^{opt} \in FEASIBLE$. As we showed above $FEASIBLE$ can be computed in $O(n^{1/\Delta^2})$ time.

Selecting the horizontal rectangles. Recall that for each rectangle $R_i \in \mathcal{H}$, its width, $a_i \in (\Delta, 1]$ was rounded up to a multiple of $\varepsilon^2\Delta$. Hence, there are at most $\alpha = 1/(\varepsilon^2\Delta)$ distinct widths, $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_\alpha$, in $\mathcal{H}$. We use $\mathcal{H}(\bar{a}_q)$ to denote the subset of $\mathcal{H}$ consisting of all rectangles with width $\bar{a}_q$. Let $\mathcal{H}' \subseteq \mathcal{H}$. We define the *profile* of $\mathcal{H}'$ as an α-tuple $(h'_1, h'_2, \ldots, h'_\alpha)$ such that each entry $h'_q \in (0,1]$ $(q = 1, \ldots, \alpha)$ is the total height of the rectangles in $\mathcal{H}' \cap \mathcal{H}(\bar{a}_q)$.

Consider the profile $(h^*_1, h^*_2, \ldots, h^*_\alpha)$ of $\mathcal{H} \cap R^{opt}$. Note that if each value h^*_i is rounded up to the nearest multiple of ε/α, this might increase the height of the frame where the rectangles are packed by at most $\alpha(\varepsilon/\alpha) = \varepsilon$. The advantage of doing this, is that the number of possible values for each entry of the profile of $\mathcal{H} \cap R^{opt}$ is only constant, i.e. α/ε, and, the total number of profiles is also constant, $\alpha^{\alpha/\varepsilon}$.

By trying all possible profiles with entries that are multiples of ε/α we ensure to find one that is identical to the rounded profile for $\mathcal{H} \cap R^{opt}$. However, the profile itself does not yield the set of rectangles in $\mathcal{H} \cap R^{opt}$. Fortunately, we do not need to find this set, since (from the algorithms in [8] it can be shown that) any set $\mathcal{H}''$ of rectangles with the same rounded profile as $\mathcal{H} \cap R^{opt}$ can be packed along with $\mathcal{L} \cap R^{opt}$ in a frame of height $1+\varepsilon$ by solving a fractional strip-packing problem:

- Assume that we know an optimal set of large rectangles $\mathcal{L} \cap R^{opt}$ and a packing for it as described in Lemma 11. This assumption can be made since the set $FEASIBLE$ has polynomial size and for each set in $FEASIBLE$ there is a constant number of possible packings with the structure defined in Lemma 11. Thus, we can try all packing for all sets in $FEASIBLE$ in polynomial time, and one of them has to be identical to the packing of $\mathcal{L} \cap R^{opt}$. Assume also that we know the profile $(h^*_1, h^*_2, \ldots, h^*_\alpha)$ of $\mathcal{H} \cap R^{opt}$.
- For this packing of $\mathcal{L} \cap R^{opt}$ trace a grid of size ε/α over the entire square frame. Each square of this grid not occupied by a large rectangle is labelled either "h" or "sv". Squares labelled "h" will be used to pack rectangles from $\mathcal{H}$ and squares labelled "sv" will be used to pack rectangles from $\mathcal{V} \cup \mathcal{S}$. Try all labellings

for the grid's squares (there is only a constant number of them); one of them must be identical to the labelling induced by an optimum packing for R^{opt}.

- Group horizontally-adjacent grid squares labelled "h" into strips.
- The fractional strip packing problem is to fractionally pack rectangles of width $\bar{a}_i$ and total height h_i^*, $1 \le i \le \alpha$, into these strips. In this fractional packing problem a rectangle can only be split into rectangles of smaller height and the same width as the original rectangle.

Let $\mathcal{H}''$ be the set of rectangles (fractionally) packed as described above. To convert this fractional packing into an integer one, the height of the strips might need to be slightly increased. The total increase in the height of the packing is at most $(\alpha/\varepsilon)\varepsilon^4\Delta = \varepsilon$. (For a more detailed explanation, the reader is referred to [8].)

Thus, we just need to find a set of rectangles from $\mathcal{H}$ with nearly-maximum profit and with the same rounded profile as $\mathcal{H} \cap R^{opt}$. We say that a subset $\mathcal{H}' \subseteq \mathcal{H}$ is *feasible* if

- each entry $h'_q \in (0,1]$ $(q = 1,\dots,\alpha)$ in the profile of $\mathcal{H}'$ is a multiple of ε/α, and
- each subset $\mathcal{H}' \cap \mathcal{H}(\bar{a}_q)$ $(q = 1,\dots,\alpha)$ is a $(1-\varepsilon)$-approximate solution of an instance of the knapsack problem where h'_q is the knapsack's capacity and each rectangle $R_i \in \mathcal{H}(\bar{a}_q)$ is an item of size b_i and profit p_i.

Lemma 12. *In $O(n^2 \cdot 1/\varepsilon)$ time we can find the set $FEASIBLE_{\mathcal{H}}$ consisting of all feasible subsets of $\mathcal{H}$.*

Proof. There are $O(1)$ possible profiles. For each entry in a profile, in order to find a $(1-\varepsilon)$-solution for the corresponding knapsack problem, we can use the FPTAS of [17] with $O(n^2 \cdot 1/\varepsilon)$ running time. □

Selecting the vertical rectangles. We use similar ideas as above to define *profiles* and to find the set $FEASIBLE_{\mathcal{V}}$ consisting of all *feasible* subsets of $\mathcal{V}$. Note that a set $\mathcal{V}'' \subseteq \mathcal{V}$ of rectangles with the same rounded profile as $\mathcal{V} \cap R^{opt}$ can be packed, along with $\mathcal{L} \cap R^{opt}$ and a set $\mathcal{H}'' \subseteq \mathcal{H}$ as described above, in a square frame of size $1+\varepsilon$. To see this, consider a grid as described above and mark in this grid the squares occupied by rectangles from $\mathcal{V} \cap R^{opt}$ in an optimum solution. The rectangles in $\mathcal{V}''$ can be placed in these marked grid squares by solving a fractional strip packing problem as described above. This time the width of the frame needs to be increased to $1+\varepsilon$.

Selecting the small rectangles. Assume that we are given feasible subsets $\mathcal{L}' \in FEASIBLE$, $\mathcal{H}' \in FEASIBLE_{\mathcal{H}}$, $\mathcal{V}' \in FEASIBLE_{\mathcal{V}}$ such that $area(\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}') \le (1+2\varepsilon)^2$ (Recall that the rounding involved in packing the rectangles in $\mathcal{H} \cup \mathcal{V}$ increases the size of the frame of Lemma 11 to $1+2\varepsilon$). A subset $\mathcal{S}' \subseteq \mathcal{S}$ is feasible for the selection $\mathcal{L}'$,$\mathcal{H}'$, $\mathcal{V}'$, if $\mathcal{S}'$ is a $(1-\varepsilon)$-approximate solution for the instance of the knapsack problem where $(1+2\varepsilon)^2 - area(\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}')$ is the knapsack's capacity, and each rectangle $R_i \in S$ is an item of size $a_i b_i$ and profit p_i.

Proposition 13. *Given sets $\mathcal{L}' \subseteq FEASIBLE$, $\mathcal{H}' \subseteq FEASIBLE_{\mathcal{H}}$, and $\mathcal{V}' \subseteq FIASIBLE_{\mathcal{V}}$, a feasible subset $\mathcal{S}'$ of $\mathcal{S}$ can be found in $O(n^2 \cdot 1/\varepsilon)$ time.*

3.2. *The Algorithm*

Algorithm W_ε:
INPUT: A set of rectangles R, accuracy $\varepsilon > 0$.
OUTPUT: A packing of a subset of R within $[0,1+3\varepsilon] \times [0,1+3\varepsilon]$.

(1) **For each** $k \in \{0,1\dots,2/\varepsilon^2-1\}$ form the group $L^{(k)}$ of rectangles $R_i \in R$ as described above and perform Steps 2 and 3.

(2) Let $\alpha = 1/(\varepsilon^3\Delta)$.
  (a) Partition $R \setminus L^{(k)}$ into sets $\mathcal{L}, \mathcal{S}, \mathcal{H}$, and $\mathcal{V}$ as described above.
  (b) Round the sizes of the rectangles $\mathcal{L} \cup \mathcal{H} \cup \mathcal{V}$ as indicated in Lemma 11.
  (c) Compute the set $FEASIBLE$ containing all subsets of $\mathcal{L}$ with at most $1/\Delta^2$ rectangles.
  (d) Compute the set $FEASIBLE_{\mathcal{H}}$ containing all *feasible* subsets of $\mathcal{H}$ with *profiles* $(h_1,h_2,\dots,h_\alpha)$ where each entry $h_q \le 1$ $(q = 1,\dots,\alpha)$ is a multiple of ε/α.
  (e) Compute the set $FEASIBLE_{\mathcal{V}}$ containing all *feasible* subsets of $\mathcal{V}$ with *profiles* $(v_1,v_2,\dots,v_\alpha)$ where each entry $v_q \le 1$ $(q = 1,\dots,\alpha)$ is a multiple of ε/α.

(3) For each set $\mathcal{L}' \in FEASIBLE$, $\mathcal{H}' \in FEASIBLE_{\mathcal{H}}$, and $\mathcal{V}' \in FEASIBLE_{\mathcal{V}}$ do:
  (a) Try all possible packings for $\mathcal{L}'$ in the frame $[0,1+\varepsilon] \times [0,1+\varepsilon]$, positioning the rectangles as indicated in Lemma 11.
  (b) For each packing of $\mathcal{L}'$ in the frame of size $1+2\varepsilon$, split the empty space with a grid of size ε/α. Try all possible labellings for the grid's squares in which a square is labelled either $\ell_{\mathcal{H}}$ of $\ell_{\mathcal{V}}$. For each labelling, try to pack the rectangles from $\mathcal{H}'$ into the grid squares labelled $\ell_{\mathcal{H}}$, and try to pack $\mathcal{V}'$ into the squares labelled $\ell_{\mathcal{V}}$ by solving a fractional strip-packing

problem as described above.

(c) If there is a packing for $\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}'$ in the frame of size $1+2\varepsilon$, find a subset $\mathcal{S}' \subseteq \mathcal{S}$ which is *feasible* for $\mathcal{L}', \mathcal{H}'$ and $\mathcal{V}'$.

(d) Increase the size of the frame to $[1+3\varepsilon] \times [1+3\varepsilon]$ and use the NFDH algorithm to pack the rectangles $\mathcal{S}'$ within the empty gaps left by $\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}'$.

(4) Among all packings computed in Step 3, output one having maximum profit.

3.3. *Proof of Theorem 1*

Lemma 14. *There exists a selection of feasible subsets* $\mathcal{L}' \in FEASIBLE, \mathcal{H}' \in FEASIBLE_{\mathcal{H}}$, $\mathcal{V}' \in FEASIBLE_{\mathcal{V}}$, *and* $\mathcal{S}' \subseteq \mathcal{S}$, *such that*

- $profit(\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}' \cup \mathcal{S}') \geq (1-\varepsilon)\mathrm{OPT}$,
- *algorithm* W_ε *outputs a packing of* $\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}' \cup \mathcal{S}'$ *within the augmented square frame* $[0,1+3\varepsilon] \times [0,1+3\varepsilon]$.

Proof. Choose $\mathcal{L}' = \mathcal{L} \cap R^{opt}$. Let $\mathcal{H}' \subseteq \mathcal{H}$ and $\mathcal{V}' \subseteq \mathcal{V}$ be sets with the same rounded profiles as $\mathcal{H} \cap R^{opt}$ and $\mathcal{V} \cap R^{opt}$ and profits at least $(1-\varepsilon)\mathit{profit}(\mathcal{H} \cap R^{opt})$ and $(1-\varepsilon)\mathit{profit}(\mathcal{V} \cap R^{opt})$ respectively. Let $\mathcal{S}' \subseteq \mathcal{S}$ be a $(1-\varepsilon)$-approximate solution of the knapsack problem with knapsack capacity $(1+2\varepsilon)^2 - \mathit{area}(\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}')$ and items $R_i \in \mathcal{S}$ of size $a_i b_i$ and profit p_i. Note that $\mathit{profit}(\mathcal{S}') \geq (1-\varepsilon)\mathit{profit}(\mathcal{S} \cap R^{opt})$ and, therefore, $\mathit{profit}(\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}' \cup \mathcal{S}') \geq (1-\varepsilon)\mathit{profit}(R^{opt})$.

Since R^{opt} can be packed into a unit size square frame and the sets $\mathcal{L}', \mathcal{H}'$, and $\mathcal{V}'$ are rounded-up sets with profits at least the profits of $R^{opt} \cap \mathcal{L}$, $R^{opt} \cap \mathcal{H}$, and $R^{opt} \cap \mathcal{V}$, then, by Lemma 11 and the discussion in Section 3.1. about the selection of $FEASIBLE_{\mathcal{H}}$ and $FEASIBLE_{\mathcal{V}}$, they can be packed into a square frame of size $[0,1+2\varepsilon] \times [0,1+2\varepsilon]$. The small rectangles in $\mathcal{S}'$ have total area $(1+2\varepsilon)^2 - \mathit{area}(\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}')$ and, thus, the NFDH algorithm can pack them in the empty gaps left by the other rectangles if we increase the size of the frame to $[0,1+3\varepsilon] \times [0,1+3\varepsilon]$. This follows from a straightforward extension of Lemma 2 to rectangles. □

Algorithm W_ε considers all values $k \in \{0,1\ldots,2/\varepsilon^2 - 1\}$. For at least one of these values it must find a group $L^{(k)}$ such that

$$profit(R^{opt} \setminus L^{(k)}) \geq (1-\varepsilon^2)\mathrm{OPT}.$$

For this group, the rest of the rectangles $R \setminus L^{(k)}$ is partitioned into sets $\mathcal{L}, \mathcal{S}, \mathcal{H}$, and $\mathcal{V}$.

By Lemma 14 there exist a selection of feasible subsets $\mathcal{L}' \in FEASIBLE, \mathcal{H}' \in FEASIBLE_{\mathcal{H}}$, $\mathcal{V}' \in FEASIBLE_{\mathcal{V}}$, and $\mathcal{S}' \subseteq \mathcal{S}$, such that

$$profit(\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}' \cup \mathcal{S}') \geq (1-\varepsilon)\mathrm{OPT},$$

and such that algorithm W_ε outputs a packing of $\mathcal{L}' \cup \mathcal{H}' \cup \mathcal{V}' \cup \mathcal{S}'$ within an augmented square frame $[0,1+3\varepsilon] \times [0,1+3\varepsilon]$. Since algorithm W_ε tries all feasible sets in $FEASIBLE$, $FEASIBLE_{\mathcal{H}}$, and $FEASIBLE_{\mathcal{V}}$, and all packings for them, W_ε must find the required solution.

All feasible subsets $FEASIBLE$, $FEASIBLE_{\mathcal{H}}$ and $FEASIBLE_{\mathcal{V}}$, can be found in $O(n^2 \cdot 1/\varepsilon)$ time. Step 3(b) of algorithm W_ε can be performed by using the algorithm for strip-packing described in [8]. This algorithm also runs in time polynomial in n. Furthermore, there is only a constant number of possible packings for any set of large rectangles from $FEASIBLE$. Hence, the overall running time of algorithm W_ε is polynomial in n for fixed ε.

Conclusions

An interesting open problem is that of finding a set $R' \subseteq R$ of rectangles with profit at least $(1-\varepsilon)\mathrm{OPT}$ and a packing for them in the unit square region $[0,1] \times [0,1]$ without augmentation. Natural extensions of our algorithm (like removing one of the large rectangles to accommodate those rectangles 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 more complex problem can be solved in polynomial time, but new techniques seem to be needed.

Acknowledgements

We thank an anonymous referee for valuable comments that improved the legibility of the paper and for bringing to our attention reference [9]. This work was supported by EU-Project AEOLUS ”Algorithmic Principles for Building Efficient Overlay Computers”, IST-015964. Work of Roberto Solis-Oba was partially supported by the Natural Sciences and Engineering Research Council of Canada grant R3050A01.

References

[1] B.S. Baker, D.J. Brown, and H.P. Katseff. A 5/4 algorithm for two dimensional packing. *Journal of*

Algorithms, 2:348–368, 1981.

[2] B.S. Baker, A.R. Calderbank, E.G. Coffman, and J.C. Lagarias. Approximation algorithms for maximizing the number of squares packed into a rectangle. *SIAM Journal on Algebraic and Discrete Methods*, 4:383–397, 1983.

[3] B.S. Baker, E.G. Coffman, and R.L. Rivest. Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9:846–855, 1980.

[4] N. Bansal and M. Sviridenko. New approximability and inapproximability results for 2-dimensional bin packing. In *Proceedings 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 189–196, 2004.

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

[6] F.R.K. Chung, M.R. Garey, and D.S. Johnson. On packing two-dimentional bins. *SIAM Journal on Algebraic and Discrete Methods*, 3:66–76, 1982.

[7] E.G.Jr. Coffman, M.R. Garey, D.S. Johnson, and R.E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9:808–826, 1980.

[8] J.R. Correa and C. Kenyon. Approximation schemes for multidimensional packing. In *Proceedings 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 179–188, 2004.

[9] D. Fayard and V. Zissimopoulos. An approximation algorithm for solving unconstrained two-dimensional knapsack problems. *European Journal of Operational Research*, 84:618–632, 1995.

[10] C.E. Ferreira, F.K. Miyazawa, and Y. Wakabayashi. Packing squares into squares. *Perquisa Operacional*, 19:349–355, 1999.

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

[12] P.C. Gilmore and R.E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13:94–120, 1965.

[13] I. Golan. Performance bounds for orthogonal, oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 10:571–582, 1981.

[14] L. A. Hall. Approximability of flow shop scheduling. *Mathematical Programming*, 82:175–190, 1998.

[15] K. Jansen and G. Zhang. Maximizing the number of packed rectangles. In *Proceedings 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 362–371, 2004.

Received 29 March, 2005; revised 23 May 2007; accepted 29 May 2007

[16] K. Jansen and G. Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47:323–342, 2007.

[17] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer, 2004.

[18] C. Kenyon and E. Rémila. A near-optimal solution to a two dimentional cutting stock problem. *Mathematics of Operations Research*, 25:645–656, 2000.

[19] D.J. Kleitman and M.M. Krieger. An optimal bound for two dimensional bin packing. In *Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 163–168, 1975.

[20] E. Lawler. Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research*, 4:339–356, 1979.

[21] J.Y.-T. Leung, T.W. Tam, C.S. Wong, G.H. Young, and F.Y.L. Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10:271–275, 1990.

[22] A. Meir and L. Moser. On packing of squares and cubes. *Journal of Combinatorial Theory*, 5:126–134, 1968.

[23] J.W. Moon and L. Moser. Some packing and covering theorems. *Colloquium Mathematicum*, 17:103–110, 1967.

[24] L. Moser. Poorly formulated unsolved problems of combinatorial geometry. Mimeographed, 1965.

[25] W. Moser and J. Pach. Problem N.108. In *Research Problems in Discrete Geometry*. McGill University, Montreal, 1989.

[26] P. Novotny. A note on packing of squares. In *Studies of University of Transport and Communications in Zilina*, number 10 in Math.-Phys. series. 1995.

[27] P. Novotny. On packing of squares into a rectangle. *Archivum Mathematicum*, 32:75–83, 1996.

[28] I. Schiermeyer. Reverse fit : a 2-optimal algorithm for packing rectangles. In *Proceedings 2nd European Symposium on Algorithms (ESA)*, pages 290–299, 1994.

[29] S. Seiden and R. van Stee. New bounds for multi-dimentional packing. *Algorithmica*, 36(3):261–293, 2003.

[30] S.V. Sevastianov and G.J. Woeginger. Makespan minimization in open shops: a polynomial time approximation scheme. *Mathematical Programming*, 82:191–198, 1998.

[31] D.D. Sleator. A 2.5 times optimal algorithm for bin packing in two dimensions. *Information Processing Letters*, 10:37–40, 1980.

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