This paper appeared in: Computational Geometry 44 (2011), 456-463

On Packing Squares into a Rectangle

Stefan Hougardy

Research Institute for Discrete Mathematics
University of Bonn
Lennéstr. 2
53113 Bonn, Germany

hougardy@or.uni-bonn.de

February 15, 2011

Abstract. We prove that every set of squares with total area 1 can be packed into a rectangle of area at most $2867/2048 = 1.399\ldots$. This improves on the previous best bound of 1.53. Also, our proof yields a linear time algorithm for finding such a packing.

1 Introduction

In 1966 Moser [15, 16] posed the following problem:

> What is the smallest number A such that every set of squares of total area 1 can be accommodated in some rectangle of area A?

Here, "accommodated" means that the rectangle and squares must be axis-parallel and no two squares intersect in their interiors. See Figure 1 for an example.

Moon and Moser [14] proved that $1.2 \leq A \leq 2$. Kleitman and Krieger [7] improved this, showing that $A \leq \sqrt{3} < 1.733$ and then later that $A \leq 4/\sqrt{6} < 1.633$ [8]. The previously best bounds are due to Novotný, who showed that $A \geq (2+\sqrt{3})/3 > 1.244$ [17] — which is easily seen by considering a square with area 1/2 and three squares each with area 1/6 — and later that $A < 1.53$ [18]. In this paper we improve on these results by showing that $A \leq 2867/2048 = 1.39990\ldots$.

Theorem 1
Any set of squares with total area 1 *can be packed into a rectangle of area* 2867/2048*.*

Table 1 summarizes the progress on upper bounds for Moser's problem. For more background, see the paper by Moser [16], and the book by Croft, Falconer, and Guy [3] or Brass, Moser, and Pach [2].

This paper appeared in: Computational Geometry 44 (2011), 456-463

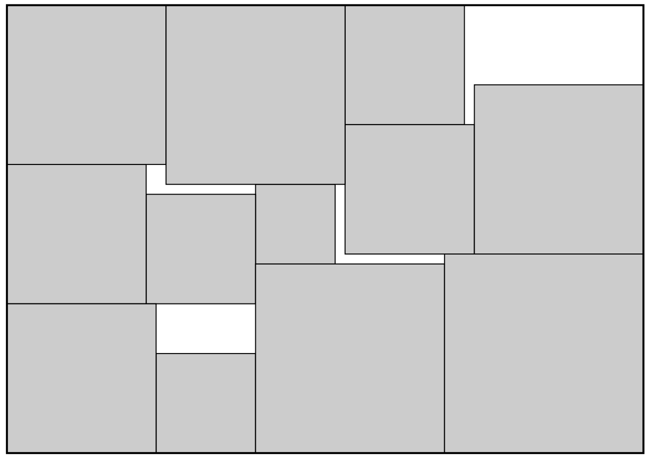

Figure 1: A packing of 12 squares into a rectangle.

authors	year	upper bound
Moon, Moser [14]	1967	2
Kleitman, Krieger [7]	1970	1.733
Kleitman, Krieger [8]	1975	1.633
Novotný [18]	1996	1.53
Hougardy (this paper)	2010	1.4

Table 1: Progress on the upper bound for the smallest area of a rectangle into which one can pack every set of squares with total area 1.

2 Outline of our Proof

In this paper we consider the *rectangle packing problem*. The input is a rectangle R with width W and height H, and a list of n rectangles $r_1, r_2, \ldots, r_i$ each with a given width w_i and height h_i. The question is whether the n rectangles can be packed into R so that no two rectangles intersect in their interiors. We do not allow rotations of the rectangles, i.e., all edges corresponding to the widths of the rectangles have to be parallel.

Our main idea is to reduce the proof of Theorem 1 to a finite number of rectangle packing problems. Kleitman and Krieger [7, 8] and also Novotný [18] take a similar approach. However, to simplify their proof they introduce dependencies among the sides of the input rectangles. We do not use such dependencies. This increases tremendously the number of cases we must consider, but also allows us to generate the cases — and thus a proof of Theorem 1 — by computer.

As we shall explain, it is challenging to write a computer program that strikes a good balance between the number of cases (several million to prove $A < 1.4$) and the running time (several weeks on a single processor).

Our program has over three thousand lines of code. At its core is an algo-

This paper appeared in: Computational Geometry 44 (2011), 456-463

rithm that solves the rectangle packing problem efficiently for instances of at most 14 rectangles.

We will not describe our program that generates the complete case distinction for the proof of Theorem 1 in full detail. Instead we provide a rough explanation of it that will suffice to understand the structure of the proof it generates. In the second part we describe a simple automated method that checks the generated proof. Thus to verify the correctness of Theorem 1 it suffices to check that our verification method — described in Section 6 — is correct and that the accompanying C++ program [5] is in turn a valid implementation of this method. Running the verification program on our proof of Theorem 1 takes only a few seconds [5].

The rest of the paper is organized as follows. In Section 3 we describe the main result we used in reducing a proof of Theorem 1 to a finite number of rectangle packing problems. In Section 4 we explain how this reduction works in theory and what ideas are needed to make it work also in practice. In Section 5 we outline our algorithm for generating the complete case distinction based on the reduction described in Section 4. As mentioned earlier, this is only a rough description, but sufficient to understand the structure of the proof. The most important part will be the description of the verification process that we present in Section 6. In Section 7 we derive some algorithmic consequences from our proof of Theorem 1. In Section 8 we describe the computational effort to obtain our proof. We end our paper with Section 9 where we discuss some potential improvements of our approach.

3 The result of Meir and Moser

Kleitman and Krieger [8] proved that every set of squares with total area 1 can be packed into a rectangle of area at most $4/\sqrt{6} < 1.633$. More precisely they proved the following stronger statement.

Theorem 2 (Kleitman, Krieger 1975)
Any set of squares with total area V can be packed into a rectangle of size $\sqrt{2 \cdot V} \times \sqrt{4 \cdot V/3}$.

As the authors state [8, p. 163], their proof is "*rather lengthy and technical.*" Moreover, they present not the whole proof but only "*a general discussion of the methods used and an outline of the major cases*". As Novotný [18] needs the result of Kleitman and Krieger for his proof that $A < 1.53$ this makes the situation a bit unsatisfying.

In order to allow for the complete and independent verification of our proof, we chose to use neither Kleitman and Krieger's nor Novotný's result, even though our proof could be slightly simpler if we did (see Section 8 for more comments on this).

The only previous result that we use is the following special case of a theorem of Meir and Moser [12]:

Theorem 3 (Meir, Moser 1968)
Any set of squares of sides $x_1 \geq x_2 \geq \ldots$ with total area V can be packed into any rectangle of size $a_1 \times a_2$ if $a_j > x_1$, $j = 1, 2$ and $x_1^2 + (a_1 - x_1)(a_2 - x_1) \geq V$.

We use this result as follows. Suppose we want to prove that every set of squares of sides $x_1 \geq x_2 \geq \ldots$ with total area 1 can be packed into a rectangle of area α.

We fix the sides $x_1 \geq x_2 \ldots \geq x_{k+1}$ for the first $k+1$ squares. Then we use Theorem 3 to obtain a family of rectangles $r_1, r_2, \ldots$ such that the squares with sides $x_{k+1}, x_{k+2}, \ldots$ and total area $1 - \sum_{i=1}^{k} x_i^2$ can be packed into each r_j. We now try to find a rectangle R of area at most α into which the squares with sides $x_1, x_2, \ldots, x_k$ and some r_j can be packed. If we succeed then we have shown that every set of squares in which the largest $k+1$ squares have sides $x_1, x_2, \ldots, x_k, x_{k+1}$ can be packed into a rectangle of area at most α.

This argument can be extended to the case where each x_i belongs to some interval, namely with $\underline{x}_i \leq x_i \leq \overline{x}_i$ for $i = 1, \ldots, k+1$. We apply Theorem 3 to the squares with sides $x_{k+1}, x_{k+2}, \ldots$ and total area at most $1 - \sum_{i=1}^{k} \underline{x}_i^2$. Again we will get a family of rectangles $r_1, r_2, \ldots$ such that the squares of sides $x_{k+1}, x_{k+2}, \ldots$ can be packed into each r_j for $\underline{x}_{k+1} \leq x_{k+1} \leq \overline{x}_{k+1}$. We now try to find a rectangle R of area at most α into which we can pack the squares of sides $\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_k$ together with one rectangle r_j. If we succeed in finding such packing for a rectangle R then we know that for all sets of squares of sides $x_1 \geq x_2 \geq \ldots$ with $\underline{x}_i \leq x_i \leq \overline{x}_i$ for $i = 1, \ldots, k+1$ a packing into a rectangle with area at most α exists.

By using a sufficiently fine discretization and a sufficiently large value for k, one can hope to reduce the proof of Theorem 1 to a finite number of rectangle packing problems each with a bounded number of rectangles. Notice that Theorem 2 alone does not suffice for such a result. Even if k is arbitrarily large and the discretization is arbitrarily small, then if all x_i are close to 0 the value $1 - \sum_{i=1}^{k} \underline{x}_i^2$ is close to 1 and Theorem 2 yields only a value for A that is arbitrarily close to 1.633.

As it turns out, it is possible to use Theorem 3 so that the described discretization approach does work: we will prove Theorem 1 by reducing it to a finite number of finite rectangle packing problems. But finding such a proof by brute force is computationally intractable: the number of cases needed in the proof of Theorem 1 would exceed 10^{24}.

In the next section we show how to avoid this combinatorial explosion via a more subtle approach that significantly reduces the number of cases.

4 Reduction to a Finite Number of Finite Packing Problems

With the help of Theorem 3 it is possible to reduce our proof of Theorem 1 to a finite number of finite rectangle packing problems. Unfortunately, even for moderately large k, e.g. $k = 12$, the discretization has to be quite fine, namely

This paper appeared in: Computational Geometry 44 (2011), 456-463

finer than $1/100$. But $k = 12$ and a discretization of $1/100$ would result in 10^{24} cases which is far beyond what can be tested in a reasonable amount of time. Smaller values of k need an even finer discretization and thus will not help. Larger values of k result in a much higher running time, as the problem of deciding whether a given set of rectangles can be packed into a given larger rectangle is NP-hard [11]. Already for $k = 10$ there exist instances of the rectangle packing problem that cannot be solved within a second by the fastest known algorithms.

Therefore, we decided to use an approach that adaptively adjusts the discretization and the value for k. Suppose we want to prove that for some fixed k and α every set of squares of sides $x_1 \geq x_2 \geq \ldots$ and with total area 1 can be packed into a rectangle of area α whenever x_i belongs to an interval $\underline{x}_i \leq x_i \leq \overline{x}_i$ for $i = 1, \ldots, k$. We apply Theorem 3 as described above. If this does not yield the desired packing then we split the problem into two subproblems by refining the discretization of the interval for the last square, i.e., we consider the two subproblems $\underline{x}_k \leq x_k \leq (\overline{x}_k + \underline{x}_k)/2$ and $(\overline{x}_k + \underline{x}_k)/2 \leq x_k \leq \overline{x}_k$ separately. We will recursively refine the discretization for x_k until we reach a maximum value for the discretization that we fix in advance. If this maximum value is reached we increase the value of k by 1, set the discretization of x_{k+1} to the whole interval, i.e., to the interval $0 \leq x_{k+1} \leq \overline{x}_k$ and continue with this problem.

By using this approach we reduce significantly the number of cases that have to be considered. For example when $\alpha = 1.43$ and the maximum discretization is $1/128$ we get $k = 10$ and need to consider $1,700,408$ cases instead of $2^{70} > 10^{21}$.

One point that we still have to discuss is how we apply Theorem 3. Suppose that we fixed the intervals $\underline{x}_i \leq x_i \leq \overline{x}_i$ for $i = 1, \ldots, k$. We apply Theorem 3 to the squares of sides $x_k, x_{k+1}, \ldots$ which have total area at most $1 - \sum_{i=1}^{k-1} \underline{x}_i^2$. For all values a_1, a_2 with $\overline{x}_k \leq a_2 \leq a_1$ and $x_k^2 + (a_1 - x_k)(a_2 - x_k) \geq 1 - \sum_{i=1}^{k-1} \underline{x}_i^2$ we know that the squares of sides $x_k, x_{k+1}, \ldots$ can be packed into a rectangle of size $a_1 \times a_2$. Of course, for a given value of a_1 we are only interested in the smallest value for a_2 such that the above inequalities are satisfied. This value of a_2 can be calculated as follows.

$$a_2 = \max_{\underline{x}_k \leq x_k \leq \overline{x}_k} x_k + \frac{\left(1 - \sum_{i=1}^{k-1} \underline{x}_i^2\right) - x_k^2}{a_1 - x_k} \tag{1}$$

The following lemma tells us how to compute a_2.

Lemma 4 *For $r \leq w$ we have*

$$\max_{l \leq x \leq r} \left\{ x + \frac{V - x^2}{w - x} \right\} = \begin{cases} r + \frac{V - r^2}{w - r} & \text{if } r \leq w - \sqrt{\frac{w^2 - V}{2}} \\ l + \frac{V - l^2}{w - l} & \text{if } l \geq w - \sqrt{\frac{w^2 - V}{2}} \\ 3w - 2\sqrt{2w^2 - 2V} & \text{otherwise} \end{cases}$$

Proof. We have

$$\begin{aligned}\left(x + \frac{V - x^2}{w - x}\right)' &= 1 - \frac{2x}{w - x} + \frac{V - x^2}{(w - x)^2} \\ &= \frac{1}{(w - x)^2} \cdot \left((w - x)^2 - 2x(w - x) + V - x^2\right) \\ &= \frac{2}{(w - x)^2} \cdot \left(x^2 - 2wx + \frac{w^2 + V}{2}\right)\end{aligned}$$

As we have $\left(x + \frac{V-x^2}{w-x}\right)'' = \frac{2(V-w^2)}{(w-x)^3}$ we see that the function $x + \frac{V-x^2}{w-x}$ attains its maximum for

$$x = w - \sqrt{\frac{w^2 - V}{2}}.$$

Thus for $l \le x \le r$ and $r \le w$ the function $x + \frac{V-x^2}{w-x}$ attains its maximum either at $x = w - \sqrt{\frac{w^2-V}{2}}$ or at the left or right border of the interval which proves the lemma. □

As we may assume that $a_2 \le a_1$ we can compute the smallest legal value of a_2 for a given value a_1 by using equation (1) and Lemma 4. We use again discretization to consider only a finite number of legal pairs (a_1, a_2). We only consider values for a_1 which yield an integer when divided by our maximum discretization factor. Together with the condition $\overline{x}_k \le a_2 \le a_1$ this implies that we have to try only a finite number of possible values for a_1.

5 Generating the Proof

In Section 4 we have described the ideas how to reduce the proof of Theorem 1 to a finite number of finite rectangle packing problems. In this section we now present an algorithm (see Algorithm 1) that, for a given value α, generates a proof that shows that every set of squares with total area one can be packed into a rectangle of area α. As a second parameter our algorithm requires the maximum allowed discretization δ, where $0 < \delta < 1$.

The proof is generated by the procedure GenerateProof. It gets as arguments the index k of the last square currently under consideration and the boundaries of the interval for the side x_k. If one can find a packing of the squares with sides $x_1, \dots, x_{k-1}$ together with an $a_1 \times a_2$ rectangle resulting from Theorem 3 within a rectangle of area α then this case is solved. Otherwise if the maximum discretization is not reached this case is split into two subcases (lines 7 and 8 of the algorithm). Otherwise the $k + 1$st square will be considered (line 5 of the algorithm). We have $x_{k+1} \le \min(r, \sqrt{1 - \sum_{i=1}^{k} \underline{x}_i^2})$ as $x_{k+1} \le x_k \le r$ and $x_{k+1}^2 + \sum_{i=1}^{k} \underline{x}_i^2 \le 1$.

In line 9 of the algorithm a discrete set of pairs (a_1, a_2) is computed that satisfies the condition of Theorem 3. The pairs in S are required to be minimal,

This paper appeared in: Computational Geometry 44 (2011), 456-463

Algorithm 1: PROOF GENERATING ALGORITHM

Input: α, maximum discretization δ
Output: A proof showing that every set of squares with total area 1 can be packed into a rectangle of area α.

1 GenerateProof (1,0,1)

procedure GenerateProof (k, l, r);

2 $\underline{x}_k := l, \overline{x}_k := r$;
3 **if** *MeirMoserFails* (k) **then**
4   **if** $r - l \leq \delta$ **then**
5   GenerateProof $(k+1, 0, \min(r, \sqrt{1 - \sum_{i=1}^{k} \underline{x}_i^2}))$
    else
7   GenerateProof $(k, l, (r+l)/2)$
8   GenerateProof $(k, (r+l)/2, r)$

function MeirMoserFails (k): Boolean

9 $S := \{$minimal pairs (a_1, a_2) with $a_1/\delta, a_2/\delta \in \mathbb{N}$ that satisfy (1)$\}$
10 **while** $S \neq \emptyset$ **do**
11   remove a pair (a_1, a_2) from S
12   **if** *squares* $\overline{x}_1, \ldots, \overline{x}_{k-1}$ *and* $a_1 \times a_2$*-rectangle can be packed into a rectangle* R *of area* α **then**
13   output proof case
14   **return** (false)
15 **return** (true)

i.e., if (a_1, a_2) is a pair in S then for all pairs (a_1, x) which satisfy equation (1) we have $a_2 \leq x$. This guarantees that the set S considered in line 9 is finite. We try to find a packing of the squares with sides $\overline{x}_1, \ldots, \overline{x}_{k-1}$ and one of the $a_1 \times a_2$ rectangles with $(a_1, a_2) \in S$ into a rectangle R of area α. If we succeed in finding such a packing we output a proof for this case (line 13). Otherwise the function MeirMoserFails returns the value true (line 15).

It remains to comment on line 12 of the algorithm. Here we have to decide whether a set of k fixed rectangles can be packed into a rectangle of area at most α. This of course is the most time consuming part of the algorithm and therefore one needs an extremely fast test for this part in order get results for values of α around 1.4 .

The rectangle packing problem is well known to be strongly NP-hard [11] and the fastest known exact algorithm for this problem has a worst case running time of $O(n!4^n/n^{1.5})$ [4]. Even though several algorithms are known that turn out to be much faster in practice [9, 13, 10, 6], these algorithms were still much too slow for our application.

Our algorithm is based on the work of Moffitt and Pollack [13]. They presented a fast algorithm to solve the packing problem which has the additional advantage that it scales, i.e., if one multiplies the widths and heights of all input rectangles by the same amount then this will not affect its running time. This property is very important to our application as we have to scale all numbers by the value $1/\delta$ to make them all integers. The algorithm of Moffitt and Pollack is on average still much too slow for our application. Therefore we added a large set of efficient heuristics to find a solution or to prove that no solution exists. While these heuristics cannot improve the worst case running time they allowed to reduce the average running time by more than a factor of 100 for the instances appearing in our proof.

6 Verifying the Proof

If a rectangle packing problem has a positive answer, then a solution can easily be specified by assuming that the rectangle R has its lower left corner in the point $(0,0)$ and its upper right corner at the point (W,H). Then it suffices to specify for each rectangle r_i a position (x_i, y_i) of its lower left corner. Feasibility of a solution then is equivalent to the two conditions

$$x_i \geq 0, x_i + w_i \leq W, y_i \geq 0, y_i + h_i \leq H \quad \forall 1 \leq i \leq n \tag{2}$$

$$\begin{aligned} x_i + w_i \leq x_j \quad \text{or} \quad x_j + w_j \leq x_i \quad \text{or} \\ y_i + h_i \leq y_j \quad \text{or} \quad y_j + h_j \leq y_i \end{aligned} \qquad \forall 1 \leq i < j \leq n \tag{3}$$

The output of the algorithm described in the previous sections for $\alpha = 1.9$ and a maximum discretization of $1/16$ is shown in Figure 2. To avoid numerical problems, all numbers in the proof are divided by the maximum discretization δ, i.e., in the example by $1/16$ so that all numbers are integers.

The syntax of the proof is as follows. The first line of the proof contains two values. The first value is 1 divided by the maximum discretization δ. The second value is the maximum area of a rectangle that is needed in the proof to pack all squares into it. As all lengths are divided by the maximum discretization one has to divide this value by the square of the maximum discretization. In our case we get $486/16^2 < 1.9$. Thus the proof shows that every set of squares with total area 1 can be packed into a rectangle of area at most 1.9 .

All lines of the proof that follow contain a single case of the proof. In each case of the proof the first k squares with sides $x_1 \geq x_2 \geq \ldots \geq x_k$ are considered where $\underline{x}_i \leq x_i \leq \overline{x}_i$ for $i = 1, \ldots, k$. The line therefore starts with the value of k followed by the k pairs of numbers $\underline{x}_i$ and $\overline{x}_i$ for $i = 1, \ldots, k$ where $\underline{x}_i$ and $\overline{x}_i$ are separated by a "-". The next two numbers, separated by an "x" denote the size of a rectangle r that satisfies Theorem 3 applied to the squares of sides $x_k \geq x_{k+1} \geq \ldots$.

```
16 486
1 0-8 22x22 [22,22] (0,0) ratio 484
2 8-9 0-4 18x17 [18,27] (0,0) (0,10) ratio 486
2 8-9 4-5 18x18 [18,27] (0,0) (0,9) ratio 486
3 8-9 5-6 0-6 18x17 [18,27] (9,0) (0,0) (0,10) ratio 486
3 8-9 6-7 0-7 18x17 [18,27] (9,0) (0,0) (0,10) ratio 486
3 8-9 7-8 0-8 17x17 [17,28] (8,0) (0,0) (0,11) ratio 476
3 8-9 8-9 0-9 16x16 [18,27] (9,0) (0,0) (0,11) ratio 486
2 9-10 0-5 18x17 [18,27] (0,0) (0,10) ratio 486
2 9-10 5-6 20x16 [30,16] (0,0) (10,0) ratio 480
3 9-10 6-7 0-7 17x16 [17,28] (7,0) (0,0) (0,12) ratio 476
3 9-10 7-8 0-8 16x16 [18,27] (8,0) (0,0) (0,11) ratio 486
3 9-10 8-9 0-9 15x15 [19,25] (9,0) (0,0) (0,10) ratio 475
3 9-10 9-10 0-10 14x14 [14,34] (0,10) (0,0) (0,20) ratio 476
2 10-11 0-5 17x16 [17,28] (0,0) (0,12) ratio 476
2 10-11 5-6 17x17 [17,28] (0,0) (0,11) ratio 476
2 10-11 6-7 19x16 [30,16] (0,0) (11,0) ratio 480
3 10-11 7-8 0-8 16x14 [19,25] (8,0) (0,0) (0,11) ratio 475
3 10-11 8-9 0-9 14x14 [14,34] (0,9) (0,0) (0,20) ratio 476
3 10-11 9-10 0-9 13x12 [13,37] (0,14) (0,0) (0,25) ratio 481
3 10-11 10-11 0-8 11x11 [11,44] (0,33) (0,0) (0,22) ratio 484
2 11-12 0-12 17x16 [17,28] (0,0) (0,12) ratio 476
2 12-13 0-11 15x15 [15,32] (0,0) (0,17) ratio 480
2 13-14 0-10 14x13 [14,34] (0,20) (0,0) ratio 476
1 14-16 23x21 [23,21] (0,0) ratio 483
```

Figure 2: An automatically generated proof that shows that every set of squares with total area 1 can be packed into a rectangle of area $486/256 < 1.9$.

The sides a and b of a rectangle R that is large enough so that the squares of sides $x_1, \dots, x_{k-1}$ and the rectangle r can be packed into it follows in the syntax $[a, b]$. Then k two-dimensional locations follow in the form "(x, y)" which denote the position of the lower left corner of the i-th square for $i = 1, \dots, k-1$ and of the rectangle r. The last two entries in a line are the word "ratio" and the area of the rectangle R.

To verify the proof the following four items should be checked:

1. The cases listed in the proof cover the whole range of possible squares.

2. The dimension of the rectangle r satisfies the condition of Theorem 3.

3. The packing given by the lower left corner of the first $k-1$ squares and the rectangle r is correct, i.e., no two rectangles intersect and all lie within the rectangle R.

4. The area of the rectangle R is as claimed at the end of the line and not larger than stated in the first line of the proof.

In what follows we describe in more detail how to check the four items. A C++ program that reads a given proof and performs all four tests is available for download [5].

To check Item 1. we further assume that the cases are ordered lexicographically. We use a variable *NextIndex* which tells us, for which x_i we expect a change between the case in the current line and the case in the next line. In the beginning *NextIndex* has the value 1 as we expect to see an interval for x_1.

We then have to check that the interval for $x_{NextIndex}$ starts with the same value as it ended in the line before.

For $i > NextIndex$ the value of $\underline{x}_i$ must be zero, as a new interval starts here. If some value of $\overline{x}_i$ is smaller than $\sqrt{1 - \sum_{j=1}^{i-1} \underline{x}_j^2}$ and smaller than $\overline{x}_{i-1}$ this means that the interval has not been completely checked. Therefore, the largest index i with $\overline{x}_i < \sqrt{1 - \sum_{j=1}^{i-1} \underline{x}_j^2}$ and $\overline{x}_i < \overline{x}_{i-1}$ is the next value for $NextIndex$. For $i = 1$ only the condition $\overline{x}_i < \sqrt{1 - \sum_{j=1}^{i-1} \underline{x}_j^2}$ needs to be satisfied.

Checking the Items 2., 3., and 4. is much easier. For Item 2. we simply apply Lemma 4. Item 3. can be verified easily by running over all pairs of different rectangles and verifying that they do not intersect (faster algorithms exist for doing this, but they are not needed for these small instances). Finally, Item 4. simply requires multiplication and comparison of two integers.

7 Algorithmic Consequences

Our proof immediately implies the following result.

Theorem 5 *Given n squares ordered by size one can find in $O(n)$ a rectangle R and a packing of these squares into R such that the area of R is at most* 1.4 *times larger than the total area of the squares.*

Proof. First scale the input so that the total area of all squares is exactly 1. Our proof of Theorem 1 consists of a constant number of cases that require the knowledge of the largest k squares (where k is at most 12). Apply the corresponding case of the proof of Theorem 1 to these squares. This gives a packing of these k squares together with a rectangle r that satisfies Theorem 3 into a rectangle R of area smaller than 1.4.

The proof of Theorem 3 given by Moon and Moser [12] is based on a linear time algorithm to find a packing whose existence is guaranteed by the theorem. Applying this algorithm to the rectangles $k+1, \ldots, n$ and the rectangle r yields the desired packing of all n squares into R in linear time.

□

The constant involved in the $O(n)$ term is rather small which makes our algorithm also useful in practice. Bansal et al. [1] present a PTAS for the related problem of approximating a smallest rectangle into which one can pack

area A	proven value	δ	# cases	k
1.50	$6144/4096 = 1.50000$	1/64	31,934	9
1.49	$6102/4096 < 1.48975$	1/64	40,947	9
1.48	$6060/4096 < 1.47950$	1/64	54,425	10
1.47	$6020/4096 < 1.46973$	1/64	71,381	11
1.46	$5980/4096 < 1.45997$	1/64	96,136	12
1.45	$5936/4096 < 1.44922$	1/64	127,807	12
1.44	$5896/4096 < 1.43946$	1/64	173,536	13
1.50	$24576/16384 = 1.50000$	1/128	143,556	7
1.49	$24412/16384 < 1.49000$	1/128	212,508	7
1.48	$24245/16384 < 1.47980$	1/128	311,544	8
1.47	$24084/16384 < 1.46998$	1/128	435,065	8
1.46	$23920/16384 < 1.45997$	1/128	614,753	9
1.45	$23754/16384 < 1.44983$	1/128	861,846	9
1.44	$23590/16384 < 1.43982$	1/128	1,222,038	9
1.43	$23427/16384 < 1.42988$	1/128	1,700,408	10
1.42	$23265/16384 < 1.41999$	1/128	2,437,097	12
1.41	$23100/16384 < 1.40992$	1/128	3,558,634	12
1.40	$22936/16384 < 1.39991$	1/128	5,365,339	12

Table 2: Computational results for different values of A and different discretizations δ.

a given set of squares (or rectangles). However, the constants involved in the running time of their algorithm are very large and in practice make it useless.

8 Proof Computation

In this section we provide some information on computational issues related to our proof of Theorem 1. Table 2 provides some information on the number of cases that were needed in our proof for different values of A and different maximum discretizations δ. The program that generates the proofs gets as input the desired value of A which is shown in the first column of the table. The second column contains the maximum area of a rectangle that is used in the proof for the given value of A. In most cases this value is slightly smaller than the input value of A. The last column of Table 2 shows the maximum number of squares that had to be considered in a single case of the proof.

It can be seen from the table that a value of 1/64 instead of 1/128 decreases the number of cases generated by a factor of more than 5, but the running time was longer by more than a factor of 2. Overall the running time was between 2 minutes for $A = 1.5$ and several weeks for $A = 1.4$ on a 2GHz single processor machine.

As mentioned in Section 3 our proof only uses Theorem 3. In addition, by making use of the results of Kleitman and Krieger [8] or of Novotný [18] one could simplify our proof. For $A = 1.5$ we could speed up our algorithm more than by a factor of 100 and the size of the proof decreased by a factor of about 10. However, for smaller values of A this effect dramatically decreases. Thus for $A < 1.4$ we observed a decrease of the size of the proof by only a few percent and the running time also stayed almost the same. We therefore decided that it is not worth to use the results of Kleitman and Krieger or of Novotný in our proof.

9 Conclusion

By reducing the proof to a finite number of rectangle packing problems, we have shown that every set of squares with total area 1 can be packed into a rectangle with area less than 1.4. We performed this reduction by creating a complicated computer program that generates the proof. However, the reader does not need to know any details about this program as in the second part we provide a simple verification program which confirms the validity of our proof. Checking the correctness of this verification program can be done by hand in less than an hour.

There is room for improvement in our work: Our approach is limited only by the running time of our proof generator and the size of the final proof. Each of these could be reduced if our rectangle packing algorithm handled non-integer instances. This would avoid integer rounding of the Meir-Moser rectangle, which wastes packing space.

We ran our algorithm on eight processors in parallel yielding a speedup of more than 7.9. Right now it seems that proof size is more of a limit to our approach than running time.

Very recently Zernisch[19] used a quadratic programming approach that allows the use of dependencies among the intervals of the x_is. The proofs generated by this method are much smaller but the running time is extremely large. Thus, currently this method only allows to prove values of A around 1.5.

References

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

[2] Peter Brass, William O.J. Moser, and János Pach. *Research Problems in Discrete Geometry*. Springer Science+Business Media, Inc., 2005.

[3] Hallard T. Croft, Kenneth J. Falconer, and Richard K. Guy. *Unsolved Problems in Geometry*. Springer Verlag, New York, 1991.

[4] Pei-Ning Guo, Chung-Kuan Cheng, and Takeshi Yoshimura. An O-tree representation of non-slicing floorplan and its applications. In *Proceedings of the 36th ACM/IEEE conference on Design automation (DAC 99)*, pages 268–273, 1999.

[5] Stefan Hougardy. On packing squares into a rectangle. `http://www.or.uni-bonn.de/~hougardy/SquarePacking.html`, January 2010.

[6] Mitsutoshi Kenmochi, Takashi Imamichi, Koji Nonobe, Mutsunori Yagiura, and Hiroshi Nagamochi. Exact algorithms for the two-dimensional strip packing problem with and without rotations. *European Journal of Operational Research*, 198:73–83, 2009.

[7] D. Kleitman and M. Krieger. Packing squares in rectangles I. *Annals of the New York Academy of Sciences*, 175:253–262, 1970.

[8] Daniel J. Kleitman and Michael M. Krieger. An optimal bound for two dimensional bin packing. In *16th Annual Symposium on Foundations of Computer Science*, pages 163–168, 1975.

[9] Richard E. Korf. Optimal rectangle packing: New results. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 142–149, 2004.

[10] N. Lesh, J. Marks, A. McMahon, and M. Mitzenmacher. Exhaustive approaches to 2D rectangular perfect packings. *Information Processing Letters*, 90:7–14, 2004.

[11] Joseph Y.-T. Leung, Tommy W. Tam, C.S.Wong, Gilbert H. Young, and Francis Y.L.Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10:271–275, 1990.

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

[13] M.D. Moffitt and M.E. Pollack. Optimal rectangle packing: a meta-csp approach. In Derek Long, Stephen F. Smith, Daniel Borrajo, and Lee McCluskey, editors, *Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling (ICAPS 2006)*, 2006.

[14] J.W. Moon and L. Moser. Some packing and covering theorems. *Colloquium Mathematicum*, 17(1):103–110, 1967.

[15] L. Moser. Poorly formulated unsolved problems of combinatorial geometry. Mimeographed, 1966.

[16] William O. J. Moser. Problems, problems, problems. *Discrete Applied Mathematics*, 31:201–225, 1991.

[17] Pavel Novotný. A note on a packing of squares. *Stud. Univ. Transp. Commun. Žilina Math.-Phys. Ser.*, 10:35–39, 1995.

[18] Pavel Novotný. On packing of squares into a rectangle. *Archivum Mathematicum (BRNO)*, 32:75–83, 1996.

[19] Jan Zernisch. Application of quadratic programming to square packing problems. Master's thesis, Research Institute for Discrete Mathematics, University of Bonn, January 2011.