The Bottom-Left Algorithm for the Strip Packing Problem

Bart Zondervan

Born November 15, 1998 in Alkmaar, The Netherlands

August 23, 2023

Master's Thesis Mathematics

Advisor: Prof. Dr. Stefan Hougardy

Second Advisor: Prof. Dr. Vera Traub

RESEARCH INSTITUTE FOR DISCRETE MATHEMATICS

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT DER

RHEINISCHEN FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

Abstract

The `Strip Packing` problem is a two-dimensional generalization of the `Bin Packing` problem. The task of the problem is to pack a set of rectangles onto a large rectangular strip such that the total height is minimized. The bottom-left algorithm packs the rectangles in the given order at the lowest possible position and if multiple lowest positions are available, then the left-most is chosen.

Building upon work of Baker et al. [BCR80] and Brown [Bro80], the lower bound for the approximation ratio of the bottom-left algorithm when using the best ordering of the rectangles is improved from $5/4$ to $4/(3+\varepsilon)$. Furthermore, in the special `Square Strip Packing` case, where all the rectangles are squares, it is improved from $12/(11+\varepsilon)$ to $4/(3+\varepsilon)$.

It is known that the approximation ratio of the bottom-left algorithm is unbounded when the rectangles are badly ordered [BCR80]. On the contrary, this thesis shows that for `Square Strip Packing` the approximation ratio is always bounded by 16. The main idea of the proof is to construct local coverings of unoccupied space by squares adjacent to it and then combining these local coverings to get the bound on the approximation ratio.

Finally, Baker et al. [BCR80] showed that the approximation ratio of the bottom-left algorithm equals 2 when squares are ordered by decreasing size. This thesis considers different orderings of the squares such that in many cases the approximation ratio becomes better-than-2. In general, it is shown that ordering the squares by increasing size results in an approximation ratio of at most 3. However, if the ratio between the width of the strip and the size of the largest square tends to infinity, then the approximation ratio approaches 2. Moreover, if the ratio between the height of the optimum packing and the size of the largest square becomes large, then the approximation ratio becomes better-than-2 as well.

Last but not least, the last-row-full ordering is studied. It is proven that for every instance, the last-row-full ordering gives a better result than the increasing size ordering. Additionally, it is shown that the last-row-full order gives a 2-approximation, and becomes better-than-2 in many cases.

Contents

1 Introduction

In the `Strip Packing` problem a rectangular strip of fixed width and infinite height is given. The task is to find an orthogonal packing of a set of rectangles into the strip such that no two rectangles overlap and the total height of the packing is minimal. Rotation of the rectangles is not allowed in this thesis.

A reduction from the `Bin Packing` problem shows that `Strip Packing` is NP-hard [Kar72]. It is even strongly NP-hard [GJ78]. Moreover, this reduction establishes that unless P=NP, there cannot exist a $(3/2 - \varepsilon)$-approximation algorithm for `Strip Packing`. Currently, the best-known approximation algorithm achieves an approximation ratio of $5/3 + \varepsilon$ [Har+14]. However, this algorithm is rather complicated and may not be of practical relevance.

In contrast, the bottom-left algorithm is extremely simple, it operates by packing the rectangles in the given order, positioning them at the lowest available point within the strip. In situations where there are multiple lowest positions, the bottom-left algorithm selects the leftmost position. An implementation of the bottom-left algorithm with quadratic time complexity is presented in the work of Chazelle [Cha83].

It is known that the bottom-left algorithm is a tight 3-approximation when the rectangles are ordered by decreasing width. Moreover, in the special `Square Strip Packing` case, where all rectangles are squares, ordering by decreasing size gives a tight 2-approximation [BCR80]. These orderings by decreasing width and size were the only constant factor approximation ratio results for the bottom-left algorithm. There are also lower bounds for the bottom-left algorithm. For example, Brown [Bro80] showed that there is an instance such that even using the best ordering of the rectangles, the approximation ratio is at least $5/4$. Similarly, Baker et al. [BCR80] showed a $12/(11 + \varepsilon)$ lower bound for `Square Strip Packing`. Thus there remain large gaps between $5/4$ and 3 for general `Strip Packing` and between $12/(11 + \varepsilon)$ and 2 for `Square Strip Packing`. The main objective of this thesis is to narrow these gaps.

1.1 Outline and contributions of this thesis

The preliminaries of this thesis are described in Chapter 2. The focus of this chapter is on formally defining the `(Square) Strip Packing` problem, introducing notation and formulating the bottom-left algorithm. Most of the work of Chapter 2 originates from Baker et al. [BCR80], but has been written in a more formal manner.

In Chapter 3 the lower bound for the bottom-left algorithm is improved. In particular, an instance is constructed in the proof of Theorem 3.1 such that the ratio between the height of the bottom-left packing using the best ordering of the instance and the height of an optimum packing is at least $4/(3+\varepsilon)$. This result improves the previously best-known lower bound of $5/4$ that was given by Brown [Bro80]. Additionally, the result is extended in Theorem 3.3 to an instance consisting of only squares that also has approximation ratio at least $4/(3 + \varepsilon)$. This improves the previously best-known lower bound of $12/(11 + \varepsilon)$ for `Square Strip Packing` instances given by Baker et al. [BCR80]. Thus this chapter is a first step in narrowing the gap between the lower and upper bound for the bottom-left algorithm.

The remaining chapters of the thesis focus on the `Square Strip Packing` problem. The main result of Chapter 4 is Theorem 4.33 showing that the approximation ratio of the bottom-left algorithm using any ordering of the squares is at most 16. This strongly contrasts the general case, there are badly ordered instances consisting of rectangles with unbounded approximation ratio [BCR80].

Proving Theorem 4.33 turns out to be rather difficult. The main idea is to show that the ratio between the unoccupied space and the occupied space is bounded by at most 12. The chapter starts in Section 4.1 and 4.2 by studying the relative position of adjacent squares that are adjacent to unoccupied space. The structure theorem (Theorem 4.13) summarizes this. Next, Section 4.3 constructs the so-called natural cover partition of unoccupied space. This cover partition enables the local cover theorems (Theorem 4.24 and 4.26) in Section 4.4 to locally cover unoccupied space by squares in the neighborhood of it. Section 4.5 takes care of unoccupied space that has not yet been covered by reducing it to different unoccupied space types (Theorem 4.30 and 4.31) or cutting off part of the top of the packing. The local cover theorems and reduction theorems are combined in Section 4.6 to define a global covering of all the unoccupied space in the packing. In particular, the global cover theorem (Theorem 4.32) shows that the ratio between the unoccupied space and the squares is at most 12. Finally, this result is used in Section 4.7 to prove Theorem 4.33 stating that the approximation ratio of the bottom-left algorithm is at most 16 regardless of the order of the squares.

The main objective of Chapter 5 is to find an ordering of the squares such that the approximation ratio of the bottom-left algorithm becomes better-than-2. Section 5.1 and 5.2 consider ordering the squares by increasing size. First, using a vertical cover theorem (Theorem 5.7), it is shown that ordering the squares by increasing size results in a 3-approximation (Theorem 5.8). After that, the analysis is improved using the so-called horizontal cover theorem (Theorem 5.11), resulting in an approximation ratio that tends to 2 as the ratio between the width of the strip and the size of the largest square goes to infinity (Theorem 5.12). It is shown in Section 5.3 that this is tight in the limit. Moreover, if the ratio between the height of the optimum packing and the size of the largest square becomes larger than 1, then the approximation ratio can become better-than-2. All in all, it turns out that in many cases, it is better to use increasing size squares in comparison to decreasing size squares as described in Baker et al. [BCR80].

In Section 5.4 a new ordering for squares is considered called last-row-full. It is shown in Theorem 5.16 that this ordering is always better than using the increasing size ordering. Moreover, Theorem 5.18 shows that the LRF-ordering gives an approximation ratio of the bottom-left algorithm equal to 2. This ratio is tight in the limit as shown in Theorem 5.19, however, for many special cases the approximation ratio becomes provable better-than-2.

1.2 Applications

Efficiently packing rectangular figures into a given rectangular area is a problem with widespread practical applications in real-world scenarios. Notable among these applications are manufacturing and logistics. For instance, in the manufacturing industry, algorithms designed to solve the `Strip Packing` problem can be used to cut rectangular pieces out of materials such as textiles, metals, and plastics. This helps to optimize the material usage and minimize production costs. Similarly, in the logistics and transportation industry, where packing space is a crucial factor, `Strip Packing` solutions can be employed to optimize the loading of cargo on ships, airplanes, or trucks. This ensures maximum space utilization and reduced transportation cost, leading to better operational efficiency.

1.3 Acknowledgements

First and foremost, I would like to thank my advisor Prof. Dr. Stefan Hougardy for his support and encouragement throughout the year. I really appreciate the opportunity to work on such an interesting topic. Furthermore, I would also like to thank Prof. Dr. Vera Traub for being the second examiner and taking the time to grade this thesis. Finally, a special thanks goes to Chen van Dam, Marc Raffelsiefen and Marena Richter for reading over parts of the thesis, asking for clarification and giving feedback.

2 Preliminaries

This chapter introduces basic concepts and notation. In particular, Section 2.1 defines the `Strip Packing` problem (SPP) and a restricted version, the `Square Strip Packing` problem (SSPP). Furthermore, in Section 2.2 the bottom-left algorithm is introduced. The majority of the definitions presented in this chapter originate from Baker et al. [BCR80] but have been rephrased here to adopt a more formal and precise structure.

2.1 Strip packing problems

The `Strip Packing` problem (SPP) is a geometric two-dimensional packing problem of a set of rectangles on a strip that has fixed width and infinite height. The objective is to minimize the total height of the packing. This section starts by defining a `Strip Packing` instance, followed by defining an orthogonal packing of an instance. After that, the height of a packing is defined, which in turn enables to formally define the optimization problem.

Definition 2.1. A `Strip Packing` *instance* $\mathcal{I}$ consists of a set rectangles $\mathcal{S} = \{S_1, \dots, S_n\}$ together with a number $W \in \mathbb{R}_{>0}$. A rectangle is a tuple $S_i = (w_i, h_i) \in (0, W] \times \mathbb{R}_{>0}$. The space $[0, W] \times \mathbb{R}_{\geq 0}$ is called the *strip*.

A `Strip Packing` instance is also called a SPP-instance. The definition of a SSP-instance basically assumes that each rectangle fits onto the strip, because the width of each rectangle is less than or equal to the width W of the strip.

For a SSP-instance, a solution is a packing of the rectangles on the strip. There are some extra constraints for the packing, such as orthogonality. The following definition formally defines a feasible packing.

Definition 2.2. Let $\mathcal{I} = (\mathcal{S}, W)$ be a SPP-instance. A *feasible* (or *orthogonal*) *packing* is a map $\sigma\colon \mathcal{S} \to 2^{[0,W]\times\mathbb{R}_{\geq 0}}$ such that

(i) Each $S = (w, h)$ is mapped to a rectangle $\sigma(S) = [x_S, x_S + w] \times [y_S, y_S + h] \subseteq [0, W] \times \mathbb{R}_{\geq 0}$.

(ii) For each pair of rectangles $S \neq S'$ the set $\sigma(S) \cap \sigma(S') \subseteq [0, W] \times \mathbb{R}_{\geq 0}$ has measure zero.

Constraint (i) of the previous definition implies that the faces of the rectangles are parallel to the axis of the strip, this is called orthogonality of the packing. Constraint (i) also implies that the rectangles are not allowed to be rotated. Furthermore, constraint (ii) states that the rectangles do not overlap, except possibly on their boundary. Figure 1 shows an example of an orthogonal packing.

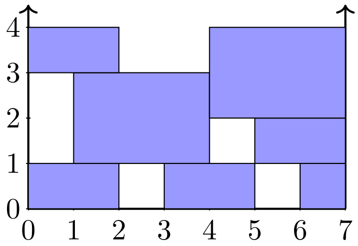


Figure 1: An orthogonal packing of $\mathcal{I} = (\{(2,1), (2,1), (1,1), (3,2), (2,1), (3,2), (2,1)\}, 7)$.

For the sake of convenience the following functions are defined to specify the x and y-coordinates of the faces of a rectangle. This will be useful for defining the height of a packing.

Definition 2.3. Let σ be an orthogonal packing for a SPP-instance $(\mathcal{S}, W)$. Let $S = (w, h) \in \mathcal{S}$ be a rectangle such that $\sigma(S) = [x_S, x_S + w] \times [y_S, y_S + h]$. The *top face*, respectively, *bottom face*, *left face* and *right face* of the rectangle S are defined by

$$\begin{aligned} \mathrm{tf}_\sigma(S) &= y_S + h, \\ \mathrm{bf}_\sigma(S) &= y_S, \\ \mathrm{lf}_\sigma(S) &= x_S, \\ \mathrm{rf}_\sigma(S) &= x_S + w. \end{aligned}$$

The top face tf_σ is often denoted by tf when the packing σ is clear from the context. The same holds for the other faces of a rectangle. Figure 2 illustrates the different faces of a rectangle.

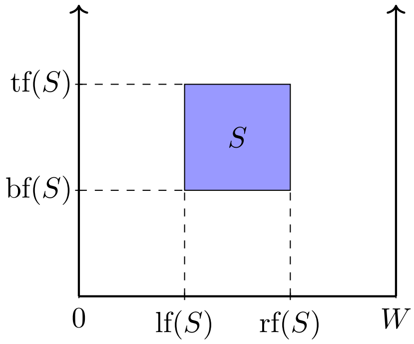


Figure 2: The top, bottom, left and right face of rectangle S.

Now, the total height of a packing is defined as the largest value of a top face of a rectangle in the packing. Ultimately, the goal is to find an orthogonal packing with minimal total height.

Definition 2.4. Let σ be an orthogonal packing for an instance $\mathcal{I} = (\mathcal{S}, W)$. Then the *total height* of the packing σ is defined by $h_\sigma(\mathcal{I}) = \max\{\mathrm{tf}_\sigma(S) \mid S \in \mathcal{S}\}$.

Often $h_\sigma(\mathcal{I})$ is written as h_σ when the instance is clear from the context. Denote by OPT an orthogonal packing of $\mathcal{I}$ such that

$$h_{\mathrm{OPT}}(\mathcal{I}) = \min\{h_\sigma(\mathcal{I}) \mid \sigma \text{ is an orthogonal packing for } \mathcal{I}\}.$$

With these definitions at hand, the `Strip Packing` problem can be formally defined in a very compact way.

STRIP PACKING PROBLEM (SPP)

Given: A SPP-instance $\mathcal{I} = (\mathcal{S}, W)$.

Task: Find an orthogonal packing σ of $\mathcal{I}$ such that the total height h_σ is minimal.

The `Bin Packing` problem (BPP) is a special case of the `Strip Packing` problem, namely it can be shown that the minimal number of bins needed for a BPP-instance $\{a_1, \dots, a_n\}$ equals the height of an optimal packing of the SPP-instance $(\{(a_1, 1), \dots, (a_n, 1)\}, 1)$. An introduction to the `Bin Packing` problem and solution methods is given in Chapter 18 of the book Combinatorial Optimization by Korte and Vygen [KV18].

The `Bin Packing` problem is NP-hard as shown by Karp [Kar72], hence it immediately follows that the `Strip Packing` problem is NP-hard. Moreover, the proof of Karp shows that there cannot exists a $(3/2-\varepsilon)$-approximation algorithm for `Bin Packing`, hence the same holds for `Strip Packing`. Furthermore, Garey and Johnson [GJ78] showed that the `Bin Packing` problem is strongly NP-hard, hence this also holds for the `Strip Packing` problem.

A special case of the `Strip Packing` problem is the `Square Strip Packing` problem (SSPP). This is exactly the same problem, but all rectangles are squares.

Definition 2.5. A `Square Strip Packing` *instance* is a SSP-instance $\mathcal{I} = (\mathcal{S}, W)$ such that for every $S \in \mathcal{S}$ it holds that $w(S) = h(S)$.

All definitions for SPP-instances also hold for SSPP-instances since it is a special case.

SQUARE STRIP PACKING PROBLEM (SSPP)

Given: A SSPP-instance $\mathcal{I} = (\mathcal{S}, W)$.

Task: Find an orthogonal packing σ of $\mathcal{I}$ such that the total height h_σ is minimal.

Leung et al. [Leu+90] proved that the `Square Strip Packing` problem is also strongly NP-hard. The proof is very different to the proofs of (strongly) NP-hardness of the `Strip Packing` problem, because there is no obvious reduction from `Bin Packing` to `Square Strip Packing`.

The remainder of this section introduces a few useful notations for the rectangles. The first such notation is for the width and height of rectangles and squares.

Definition 2.6. For a rectangle $S = (w, h)$ denote by $w(S) = w$ the *width* and by $h(S) = h$ the *height*. For a square $S = (w, h)$ it holds that $w = h$, therefore, the value of $w(S)$ and $h(S)$ is often denoted by S.

Another notation that is often used is the largest and smallest height of a rectangle in an instance. Particularly Chapter 5 uses these notations in the context of squares.

Definition 2.7. For a SPP-instance $\mathcal{I} = (\mathcal{S}, W)$, define $h_{\max} = \max\{h(S) \mid S \in \mathcal{S}\}$. Similarly, define $h_{\min}$ to be the smallest height of a rectangles in the instance.

2.2 The bottom-left algorithm

The bottom-left algorithm was first introduced by Baker et al. [BCR80]. It is a very simple algorithm for the `Strip Packing` problem that packs the rectangles in the given order as low as possible and if multiple lowest positions are available, then it chooses the left-most position. The fact that the bottom-left algorithm depends on the order of the given rectangles is important. For different orders it gives different packings, hence the following definition.

Definition 2.8. Let $\mathcal{I} = (\mathcal{S}, W)$ be a SPP-instance. Let $\mathcal{A}$ be an algorithm that given a set of rectangles returns a linear ordering of the rectangles. The *SPP-instance $\mathcal{I}$ ordered by $\mathcal{A}$* is the tuple $\mathcal{I}_\mathcal{A} = (\mathcal{S}_\mathcal{A}, W)$ where $\mathcal{S}_\mathcal{A}$ is a tuple $(S_1, \dots, S_n) \in \mathcal{S}^n$ consisting of $n = |\mathcal{S}|$ different squares ordered using $\mathcal{A}$.

An example of a polynomial time ordering algorithm is $\mathcal{A} = \text{incr}$ that orders the rectangles of an instance by increasing area. This particular ordering will be studied in Chapter 5. When the rectangles have an ordering, the bottom-left algorithm can pack them onto the strip.

Definition 2.9. Let $\mathcal{I}_\mathcal{A} = (\mathcal{S}_\mathcal{A}, W)$ be a SPP-instance ordered by $\mathcal{A}$ where $\mathcal{S}_\mathcal{A} = (S_1, \dots, S_n)$. The *bottom-left algorithm* is an algorithm that returns a packing $\text{BL}(\mathcal{I})$ that is inductively defined in the order of $\mathcal{A}$ by

$$\text{BL}(\mathcal{I})(S_i) = [x, x + w_i] \times [y, y + h_i] \subseteq ([0, W] \times \mathbb{R}_{\geq 0}) \setminus \{\text{BL}(\mathcal{I})(S_j) \mid 1 \leq j < i\}$$

where first $y \in \mathbb{R}_{\geq 0}$ is chosen as small as possible, and then $x \in [0, W]$ is chosen as small as possible such that $\mathrm{BL}(\mathcal{I})$ is an orthogonal packing. Algorithm 1 is pseudocode for an implementation of the bottom-left algorithm.

Algorithm 1 Bottom-left algorithm

```
Input: SPP-instance $\mathcal{I}_\mathcal{A} = (\mathcal{S}_\mathcal{A}, W)$ ordered by $\mathcal{A}$
Output: the bottom-left packing $\mathrm{BL}(\mathcal{I})$ of $\mathcal{S}_\mathcal{A}$ in $[0, W] \times \mathbb{R}_{\geq 0}$
for $S$ in $\mathcal{S}_\mathcal{A}$ do
    place $S$ as low as possible (at bottom)
    if there are multiple options then
        place $S$ as left as possible
    end if
end for
return the packing
```

The orthogonal packing from Figure 1 is not a bottom-left packing, since multiple rectangles can simply be placed more to the left. The bottom-left packing of the instance from this figure is depicted in Figure 3.

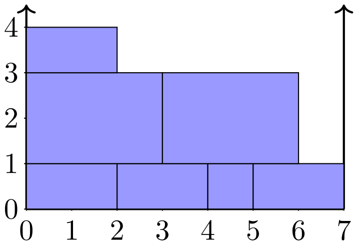


Figure 3: The bottom-left packing of $\mathcal{I} = (\{(2,1),(2,1),(1,1),(3,2),(2,1),(3,2),(2,1)\}, 7)$ with total height $h_{\mathrm{BL}}(\mathcal{I}) = 4$.

The main goal of this thesis is to study the approximation ratio of the bottom-left algorithm in different settings.

Definition 2.10. Denote by $h^{\mathcal{A}}_{\mathrm{BL}}(\mathcal{I})$ the height of the bottom-left packing of $\mathcal{I}$ ordered by $\mathcal{A}$. The *approximation ratio* of the bottom-left algorithm for the ordering $\mathcal{A}$ is defined as

$$\sup_{\mathcal{I}} \frac{h^{\mathcal{A}}_{\mathrm{BL}}(\mathcal{I})}{h_{\mathrm{OPT}}(\mathcal{I})}.$$

The height $h^{\mathcal{A}}_{\mathrm{BL}}(\mathcal{I})$ is sometimes abbreviated by $h^{\mathcal{A}}_{\mathrm{BL}}$ or h_{BL} if the instance respectively ordering is clear from the context.

The definition of an instance $\mathcal{I}_\mathcal{A}$ ordered by a sorting algorithm does not require $\mathcal{A}$ to be a polynomial time algorithm. An important algorithm of which the runtime is unknown is $\mathcal{A} = \text{best}$. For this algorithm, $\mathcal{I}_{\text{best}}$ is the ordering such that the bottom-left algorithm returns a bottom-left packing of smallest height. There are $n!$ different orderings for an instance of size n, and there does not seem to be a better way to find $\mathcal{I}_{\text{best}}$ than trying all $n!$ orderings. Another important sorting algorithm is $\mathcal{A} = \text{worst}$. Similar, $\mathcal{I}_{\text{worst}}$ is the ordering such that the bottom-left algorithm returns a bottom-left packing of largest height.

Last of all, being bottom-left defines a linear ordering on the squares that is not necesarilly the same as the linear ordering of the sorting algorithm.

Definition 2.11. Let $\mathrm{BL}(\mathcal{I}_{\mathcal{A}})$ be a bottom-left packing. Consider two rectangles $S, S' \in \mathcal{S}$. Then S is *more bottom-left than* S' if either $\mathrm{bf}(S) < \mathrm{bf}(S')$, or $\mathrm{bf}(S) = \mathrm{bf}(S')$ and $\mathrm{lf}(S) < \mathrm{lf}(S')$. This is denoted by $S <_{\mathrm{BL}} S'$.

It is also possible to compare two subsets of the strip using the definition above. In that case, the bottom face is the lowest point of the subset and the left face is the left-most point on the height of the bottom face.

3 An improved lower bound for the best bottom-left packing

Even for the best possible ordering of rectangles, the bottom-left algorithm might produce a non-optimal packing. Baker, Coffman and Rivest [BCR80] were the first to show that for any sufficiently small $\varepsilon > 0$, there exists a `Square Strip Packing` instance $\mathcal{I}_\varepsilon$ such that

$$\frac{h_{\mathrm{BL}}^{\mathrm{best}}(\mathcal{I}_\varepsilon)}{h_{\mathrm{OPT}}(\mathcal{I}_\varepsilon)} = \frac{12}{11+\varepsilon}.$$

Later an improvement was given by Brown [Bro80], showing that there exists a set of rectangles with ratio equal to 5/4. Up to now, this was the best-known lower bound for the ratio between the height of a best bottom-left packing and the height of an optimal packing. Furthermore, Baker et al. [BCR80] was the best-known lower bound for SSPP-instances.

Section 3.1 improves the lower bound for the bottom-left algorithm from 5/4 to $4/(3+\varepsilon)$ by constructing a SPP-instance and proving the ratio with the same arguments as Brown [Bro80]. After that, Section 3.2 extends the result to very large SSPP-instances to improve the former best lower bound of $12/(11+\varepsilon)$ to $4/(3+\varepsilon)$. This section requires novel arguments.

3.1 Rectangular case

The main idea in this section to show a better lower bound for the best bottom-left algorithm is to construct an instance that has an optimum packing with bottom-left structure such that the optimum packing is unique up to symmetries. After that, the instance is slightly modified, preserving the uniqueness (up to symmetry) of the optimum packing, but loosing the bottom-left structure for the optimum packing. This will result in the desired lower bound.

Theorem 3.1. For $\varepsilon > 0$, there exists a set of rectangles $\mathcal{I}_\varepsilon$ such that

$$\frac{h_{\mathrm{BL}}^{\mathrm{best}}(\mathcal{I}_\varepsilon)}{h_{\mathrm{OPT}}(\mathcal{I}_\varepsilon)} = \frac{4}{3+\varepsilon}.$$

Proof. Consider an instance with rectangles $(3,2),(3,2),(2,1),(2,1),(2,1),(2,1),(1,1)$ and strip width 7. The packings in Figure 4 of this instance with height 3 are tight, i.e., all space is occupied. Hence these packings are optimal.

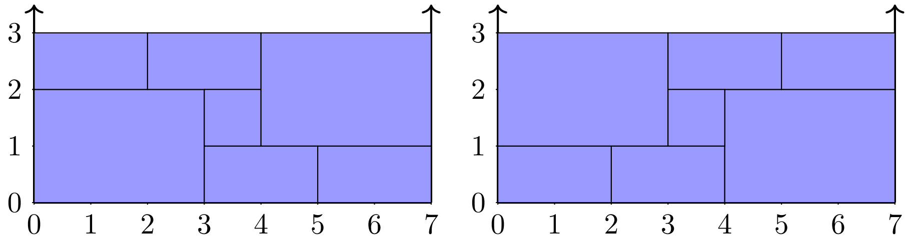


Figure 4: Optimal packings.

Claim: The packings from Figure 4 are the only optimal packings. To prove this claim, consider the three disjoint 1×7 horizontal rows for an arbitrary optimal packing. Let the type of a row be a multiset of the sizes of rectangles that the row intersects. There are three possible types: (a) $\{3,3,1\}$, (b) $\{3,2,2\}$ and (c) $\{2,2,2,1\}$.

Let a denote the number of rows of type (a), etc. There is a total of three rows, therefore it holds that $a+b+c=3$. Furthermore, there is exactly one rectangle of width 1 and this rectangle has height 1, hence it holds that $a+c=1$, thus it follows that $b=2$. Now, there are four rectangles of width 2, these rectangles all have height 1, so $2b + 3c = 4$, this implies that $c = 0$. In conclusion, required is that $a = 1$, $b = 2$ and $c = 0$.

Both rectangles of width 3 have height 2, hence the row of type (a) must be in the middle, as otherwise there must be another row of type (a), contradicting that $a = 1$. Furthermore, if the square of width 1 is placed at one of the sides of the strip, then either in the top or bottom row another square of width 1 is needed. As there is no other rectangle of width 1, the square of width 1 must be placed in the middle. Thus the packings in Figure 4 are the only possible packings with one row of type (a) and two rows of type (b). This proves the claim.

Next, the instance is modified slightly so that the two rectangles of width 3 become a bit thinner, and the square of size 1 becomes a bit higher. More formally, for $\varepsilon > 0$ sufficiently small, let $(3-\varepsilon, 2), (3-\varepsilon, 2), (2, 1), (2, 1), (2, 1), (2, 1), (1, 1+\varepsilon)$ be the rectangles of the modified instance $\mathcal{I}_\varepsilon$. Let the width of the strip still be 7. The preceding proof shows that (to within ε) the packings of Figure 4 are still optimum. However, for each of the optimal solution, the packing cannot be a bottom-left packing.

Consider the first packing of Figure 4, the two top rectangles of size $(2, 1)$ are packed last in the bottom-left algorithm. However, as the $(3, 2)$ rectangle shrinks to size $(3-\varepsilon, 2)$, there is no space to fit two $(2, 1)$ rectangles, unless if the top right $(3-\varepsilon, 2)$ is shifted a bit to the right, breaking the bottom-left structure. This packing is depicted in Figure 5.

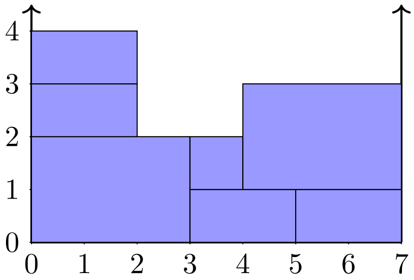


Figure 5: The bottom-left packing of $\mathcal{I}_\varepsilon$ demonstrating that there is not enough space for the two top rectangles of size $(2, 1)$.

Next, consider the second packing of Figure 4. If the rectangle of size $(1, 1+\varepsilon)$ is packed before the two top right rectangles of size $(2, 1)$, then the bottom-left algorithm returns the first packing of Figure 6. Else if one of the rectangles of size $(2, 1)$ is placed before the rectangle of size $(1, 1+\varepsilon)$, then there is no space for the rectangle of size $(1, 1+\varepsilon)$ left, hence this one will be placed on top as demonstrated in the second packing of Figure 6.

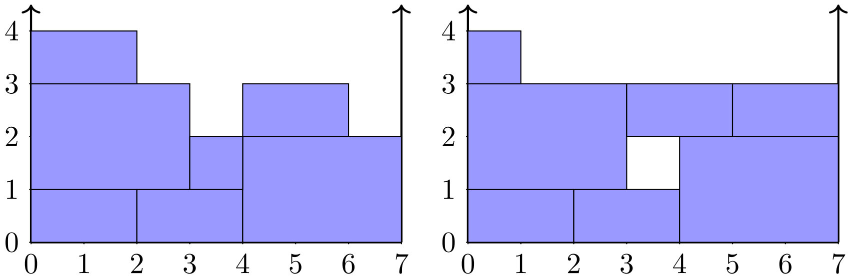


Figure 6: The bottom-left packing if the rectangle of size $(1, 1+\varepsilon)$ is placed before the two top rectangles of size $(2, 1)$, respectively, the bottom-left packing if the two top rectangles of size $(2, 1)$ are placed before the rectangle of size $(1, 1+\varepsilon)$.

In conclusion, the best bottom-left packing has height 4. Thus the modified instance has ratio between the height of the best bottom-left packing and the height of an optimal packing equal to $4/(3+\varepsilon)$. □

3.2 Square case

There is an easy modification of the instance from Theorem 3.1 such that all rectangles are squares. In Corollary 3.2 it is shown that this results in a $\frac{6-4\varepsilon}{5+\varepsilon}$ lower bound for the bottom-left algorithm for SSPP-instances. After that, the corollary is generalized to get a $\frac{4}{3+\varepsilon}$ lower bound for the bottom-left algorithm for SSPP-instances.

Corollary 3.2. For sufficiently small $\varepsilon > 0$, there exists a set of squares $\mathcal{I}_\varepsilon$ such that

$$\frac{h_{\mathrm{BL}}^{\mathrm{best}}(\mathcal{I}_\varepsilon)}{h_{\mathrm{OPT}}(\mathcal{I}_\varepsilon)} = \frac{6-4\varepsilon}{5+\varepsilon}.$$

Proof. Consider the collection of squares of size $3, 3, 2, 2, 2, 2, 1$. Let the width of the strip be 7. Notice that the widths of these squares are the same as the widths of the rectangles from the instance in the proof of Theorem 3.1. Using similar reasoning as Theorem 3.1, the two packings of Figure 7 are the only optimal packings.

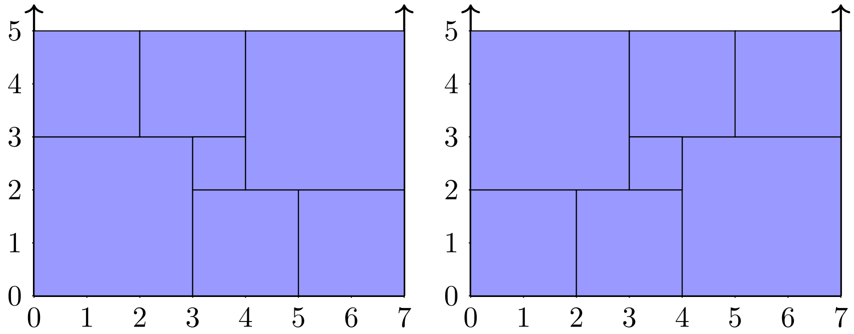


Figure 7: Optimal packings of instance consisting of only squares.

Let $\varepsilon > 0$, and consider the modified instance where the squares of size 3 get size $3-2\varepsilon$, and the square of size 1 gets size $1+\varepsilon$. Like in Theorem 3.1, the packings from Figure 7 are still optimum, but do not have the bottom-left structure.

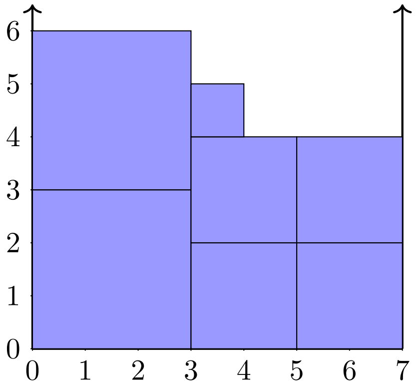


Figure 8: Best bottom-left packing for modified instance consisting of only squares.

If the squares of size $3 - 2\varepsilon$ are on top of each other, then the packing has height at least $6 - 4\varepsilon$, and as Figure 8 shows there is a bottom-left packing with height $6 - 4\varepsilon$. If the squares of size $3 - 2\varepsilon$ are not on top of each other, then a bottom-left packing has height at least $6 - \varepsilon$. Namely, suppose the height of the packing is less than $6 - \varepsilon$, then each column containing a square of size $3 - 2\varepsilon$ contains at most one other square of size $1 + \varepsilon$ or size 2. Furthermore, the squares of size $3 - 2\varepsilon$ are either adjacent or non-adjacent. If these squares are adjacent, then there must be a column with a square of size $3 - 2\varepsilon$ and two squares of size 2, hence the height of the packing is not less than $6 - \varepsilon$. Otherwise, if the squares of size $3 - 2\varepsilon$ are not adjacent, then the packing must be as in Figure 7, however, this is not a bottom-left packing. Thus if the two squares of size $3 - 2\varepsilon$ are not on top of each other, then the height of a bottom-left packing is at least $6 - \varepsilon$. Notice that this corresponds to a packing of Figure 7 where the square of size $1 + \varepsilon$ is placed last on top of the packing. Thus the best bottom-left height is $6 - 4\varepsilon$. Therefore the ratio equals $(6 - 4\varepsilon)/(5 + \varepsilon)$. ☐

Next Theorem 3.3 will improve the $\frac{6-4\varepsilon}{5+\varepsilon}$ lower bound by making the instance larger. The width of the instance will be determined by an algebraic equation that has an unique solution such that the optimum packing is tight, i.e., all space is occupied. After constructing the larger instance, the proof continues similar to the proof of Theorem 3.1. The resulting lower bound is equal to $4/(3 + \varepsilon)$.

Theorem 3.3. For sufficiently small $\varepsilon > 0$, there exists a set of squares $\mathcal{I}_\varepsilon$ such that

$$\frac{h_{\mathrm{BL}}^{\mathrm{best}}(\mathcal{I}_\varepsilon)}{h_{\mathrm{OPT}}(\mathcal{I}_\varepsilon)} \geq \frac{4}{3+\varepsilon}.$$

Proof. Let $2h + 1$ be a prime number. Consider the equation $(h + 1)n + 1 \equiv 0 \bmod 2h + 1$. As $2h + 1$ is prime, there is a solution $1 \leq n < 2h + 1$. Consider the instance consisting of one square of size h, $2(n + 1)$ squares of size $h + 1$ and $\frac{2n(h+1)+1}{2h+1}$ squares of size $2h + 1$. Observe that $\frac{n(h+1)+1}{2h+1}$ is a positive integer by the congruence from above. Let the width of the strip be equal to $(2n + 1)(h + 1) + 1$.

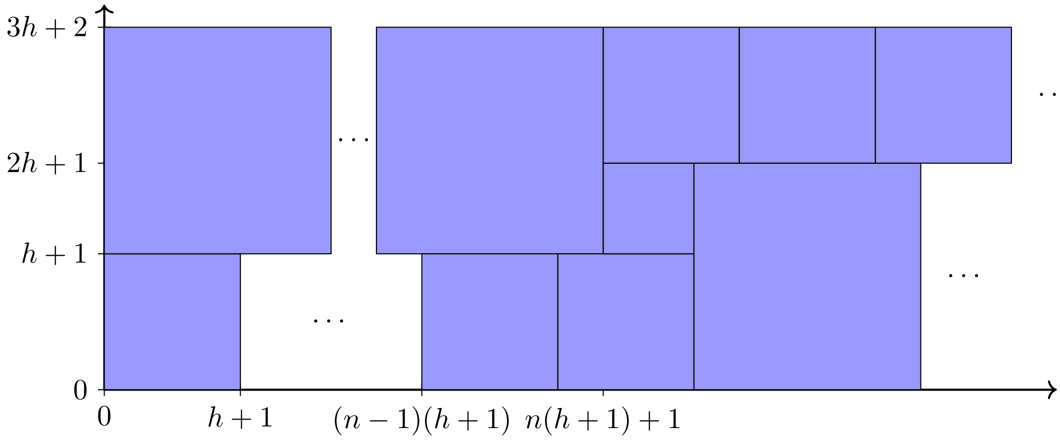
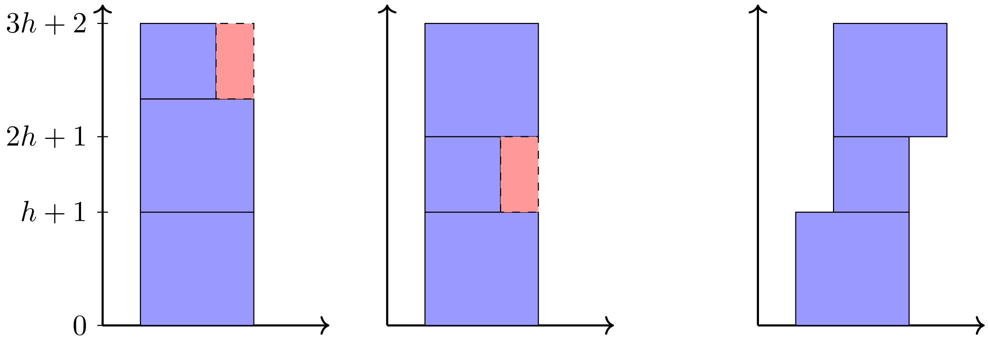


Figure 9: Unique optimal packing up to symmetry.

The packing in Figure 9 uses $n+1$ squares of size $h+1$ on the bottom left, $\frac{n(h+1)+1}{2h+1}$ squares of size $2h + 1$ on the top left, and then conversely $n + 1$ squares of size $h + 1$ on the top right, and $\frac{n(h+1)+1}{2h+1}$ squares of size $2h + 1$ on the bottom right. By definition of n this leaves a gap in the middle for the square of size h. Thus the packing is tight and hence the optimal packing has height $3h+2$.

Consider the $(2n+1)(h+1)+1$ disjoint $(3h+2)\times 1$ vertical columns. There are two possible types (a) $\{h+1, h+1, h\}$ and (b) $\{2h+1, h+1\}$. There are $(2n+1)(h+1)+1$ columns, hence it holds that $a+b=(2n+1)(h+1)+1$. Furthermore, there is one square of size h, thus $a=h$. This implies that $b=2n(h+1)+2$.

From $a=h$ it follows that either the square of size h is above (symmetrically below) two rows of squares of size $h+1$ or it is between squares of size $h+1$. The first case is not possible, as this creates a gap of height h which cannot be filled by another square, this is demonstrated by the red area in Figure 10a. In the second case, either the two squares of size $h+1$ above and below the square of size h go over the same side of the square of size h as in Figure 10b, then no other square can fill the red space, hence this is not possible. Thus the square above and below the square of size h must go in different directions over the left respectively right boundary of the square of size h, as depicted in Figure 10c.



(a) (b) (c)

Figure 10: Figure 10a show that if the square of size h is above two rows of squares of size $h+1$, then there is no other square that can fill the red area. Figure 10b shows that if the square of size h is between squares of size $h+1$, then there is no other square that can fill the red area. The only possibility where the square of size h is between squares of size $h+1$ is (up to symmetry) Figure 10c.

As $b=2n(h+1)+2$, it follows that each other column with a square of size $h+1$ also contains a square of size $2h+1$. Thus the structure in Figure 11 must be part of the optimal solution.

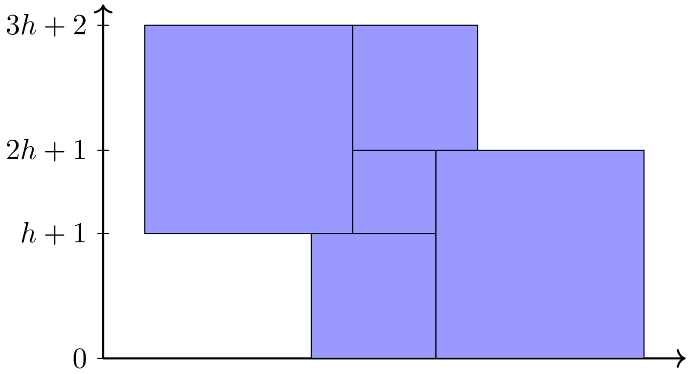


Figure 11: Up to symmetry this must be part of the optimal packing.

Now, building on the structure in Figure 11, $2n(h+1)$ more columns of type (b) are required. By definition, the number n is the smallest positive number such that the congruence equation $(h+1)n+1 \equiv 0 \bmod 2h+1$ holds. As the left square of size $2h+1$ shares already one column with a square of size $h+1$, there have to be at least $\frac{n(h+1)+1}{2h+1}$ squares of size $2h+1$ in the upper row, and at least $n+1$ squares of size $h+1$ in the lower row, to get a block that is fully occupied. The same holds for the right side of the structure in Figure 11. As there are only $\frac{2n(h+1)+1}{2h+1}$ squares of size $2h+1$ and $2(n+1)$ squares of size $h+1$, it follows that the packing in Figure 9 (up to symmetry) is the unique optimal packing.

Let $\varepsilon > 0$ be sufficienctly small and consider the slight modification where the squares of size $2h+1$ have size $2h+1-\varepsilon$ and the square of size h has size $h+\varepsilon$. The packing of Figure 9 (up to symmetry) still is the unique optimal packing, but it does not have the bottom-left structure.

Now, similar to Corollary 3.2, the packing in Figure 12 is optimal and has height $4h+2-2\varepsilon$. Because if the squares of size $2h+1-\varepsilon$ are not on top of each other, then the height of a bottom-left packing is at least $4h+2$, because there is a column containing the squares of size $2h+1-\varepsilon$, $h+1$ and $h+\varepsilon$.

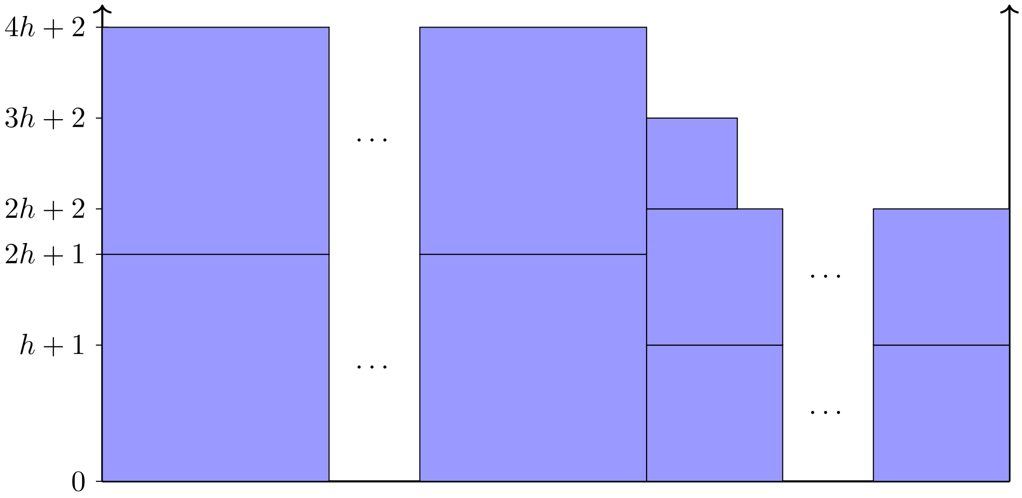


Figure 12: Best bottom-left packing.

Thus for sufficiently small $\varepsilon > 0$, and every h such that $2h+1$ is prime, there exists an instance $\mathcal{I}_{\varepsilon,h}$ such that

$$\frac{h_{\mathrm{BL}}^{\mathrm{best}}(\mathcal{I}_{\varepsilon,h})}{h_{\mathrm{OPT}}(\mathcal{I}_{\varepsilon,h})} = \frac{4h-2\varepsilon+2}{3h+\varepsilon}.$$

As there are infinitely many odd prime numbers, there always exists a SSPP-instance with the desired lower bound of $4/(3+\varepsilon)$. □

Consequently, even when 90-degree rotation of rectangles is allowed, there exists an instance such that the ratio between the height of a bottom-left packing using the best ordering and the height of an optimal solution is at least $4/(3+\varepsilon)$. Namely, the instance from Theorem 3.3 is a set of rectangles that is invariant under rotation, because each rectangle is a square.

4 Worst-case bottom-left packing

In case the rectangles are badly ordered, the ratio between the height of a bottom-left packing and the height of an optimal packing might be arbitrarily large. For example, Baker, Coffman and Rivest [BCR80] showed that there is a set of rectangles that has arbitrarily large ratio when ordered by increasing width.

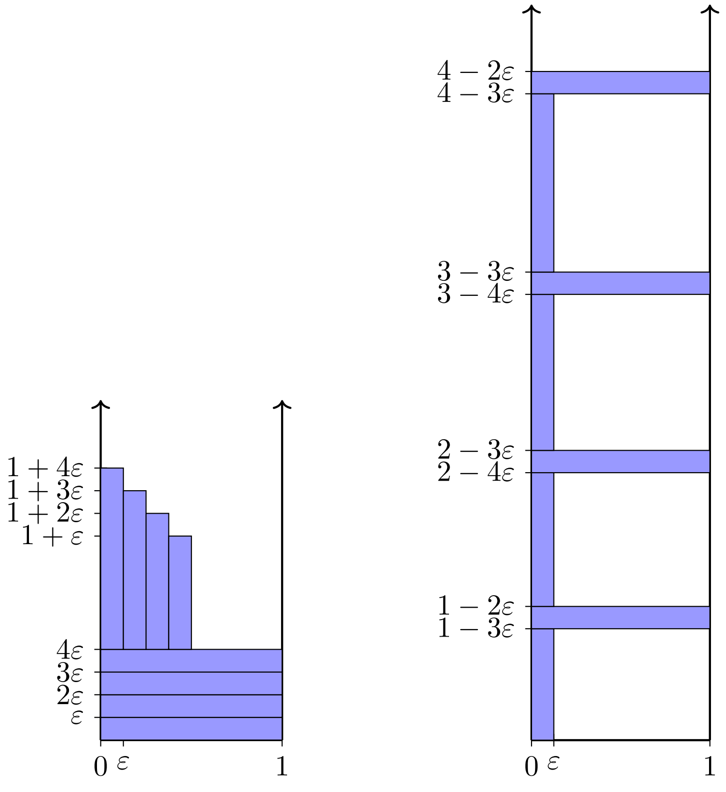
The following theorem also shows that the ratio of a badly ordered set of rectangles can be arbitrarily large, however, the construction is considerably easier. Recall that $h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I})$ denotes the height of the bottom-left algorithm using the worst ordering of $\mathcal{I}$.

Theorem 4.1. For any $M > 0$, there exists a set of rectangles $\mathcal{I}_M$ such that

$$\frac{h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I}_M)}{h_{\mathrm{OPT}}(\mathcal{I}_M)} \geq M.$$

Proof. Let M be a natural number and $0 < \varepsilon \leq \frac{1}{M}$. Consider the SPP-instance consisting of rectangles of sizes $(\varepsilon, 1 - i\varepsilon)$ for $i = 0, 1, \ldots, M - 1$ together with M rectangles of size $(1, \varepsilon)$. Let the strip have width $W = 1$.

An optimal packing has height $M\varepsilon + 1$, as the width of the rectangles of size $(1, \varepsilon)$ equals the width of the strip, hence these rectangles cannot be placed next to something else. Furthermore, at least the rectangle of size $(\varepsilon, 1)$ has to be placed somewhere. Figure 13a shows a feasible packing satisfying these two conditions, namely, all M rectangles of size $(1, \varepsilon)$ are stacked on top of each other, and the rectangles of the form $(\varepsilon, 1 - i\varepsilon)$ are all next to each other.



(a) (b)

Figure 13: Figure 13a shows an optimal packing of height $M\varepsilon + 1$. Figure 13b shows a bottom-left packing with the order described below.

Next consider the ordering where the rectangles of the form $(\varepsilon, 1 - i\varepsilon)$ are ordered by increasing height and are alternatingly followed by a rectangle of size $(1, \varepsilon)$. That is, consider the ordering $(\varepsilon, 1 - (M-1)\varepsilon), (1, \varepsilon), \ldots, (1, \varepsilon), (1 - 2\varepsilon), (1, \varepsilon), (\varepsilon, 1 - \varepsilon), (1, \varepsilon), (\varepsilon, 1)$. As seen in Figure 13b, the height of a bottom-left packing using the worst ordering of the rectangles is at least $M + \sum_{i=0}^{M-1}(1 - i\varepsilon) = 2M - \varepsilon\frac{(M-1)M}{2}$. In conclusion, for small enough $\varepsilon > 0$ it holds that

$$\frac{h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I}_M)}{h_{\mathrm{OPT}}(\mathcal{I}_M)} \geq \frac{2M - \varepsilon\frac{(M-1)M}{2}}{M\varepsilon + 1} \geq M.$$

□

The main difference between the construction of Baker et al. [BCR80] and Theorem 4.1 is that Baker et al. create a lot of unoccupied space relative to the occupied space by going very wide, while Theorem 4.1 creates a lot of unoccupied space relative to occupied space by going high and keeping the width of strip relatively small.

A previously unanswered question is whether the ratio $h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I})/h_{\mathrm{OPT}}(\mathcal{I})$ can also be arbitrarily large when the instance $\mathcal{I}$ consists of squares only. The answer turns out to be no, for every instance of the `Square Strip Packing` problem, the ratio between the height of the bottom-left packing using the worst ordering of the squares and the height of the optimal packing is smaller than a constant. The remainder of this chapter is dedicated to proving that such a constant exists. While it might be interesting to find a constant that is as small as possible, this will not be the objective of this chapter.

The main strategy, to prove that the approximation ratio is bounded, is to cover the unoccupied space in a bottom-left packing of a SSPP-instance by a fixed number of copies of the squares. This will imply that the ratio between the height of the bottom-left packing and the height of an optimal packing is bounded. To find a covering of all the unoccupied space in a packing, it is first necessary to study the structure of the bottom-left packing. More precisely, the relation between the relative positions of adjacent squares that form the boundary of unoccupied space is studied.

Instead of looking at a concrete packing, often the packing is replaced by an abstract representation of the packing in the form of a digraph. Section 4.1 defines this so-called adjacency graph which is the bedrock of the language for this chapter. Next, the core result of Section 4.2 is the structure theorem (Theorem 4.13) describing the relative relation between squares that form the boundary of unoccupied space. The structure theorem is used to construct a partition of unoccupied space. The cover partition is the subject of Section 4.3. In Section 4.4, the local cover theorem (Theorem 4.24) uses the cover partition together with the structure theorem to locally cover unoccupied space by squares that form its boundary. Section 4.5 studies unoccupied space that is not bounded by squares and also describes local coverings of these so-called trenches. It turns out that all these local coverings are compatible with each other, in the sense that there is no square that is used in arbitrarily many local coverings. Hence, there exists a constant f, such that the unoccupied space can be covered by at most f copies of the squares. The compatibility of the local coverings is demonstrated in the global cover theorem (Theorem 4.32) in Section 4.6. Finally, Section 4.7 brings everything together and shows that the approximation ratio $h_{\mathrm{BL}}^{\mathrm{worst}}/h_{\mathrm{OPT}}$ is bounded for SSPP-instances.

4.1 The adjacency graph

This section introduces the language of the structure theorem. First, the unoccupied space is partitioned in different categories. Next, the relation between the squares forming the boundary of unoccupied space is studied by defining the so-called adjacency graph. This digraph induces different types of arrows. These arrow types play a crucial role in the formulation of the structure theorem (Theorem 4.13) in Section 4.2. This chapter assumes that the reader has some background in basic graph theory. As a reference see the first few chapters of the book Combinatorial Optimization by Korte and Vygen [KV18].

In easy terms, the unoccupied space is all the space in the substrip $[0, W] \times [0, h_{\mathrm{BL}}]$ of a bottom-left packing that is not occupied by squares. The following definition describes connected unoccupied space that gets bounded after placing the i-th square from the instance. Although some parts of this unoccupied space might be filled by squares later, in the sequel it will be useful to have considered the unoccupied space and its surrounding squares at the moment it gets bounded.

Definition 4.2. Let BL be a bottom-left packing of $\mathcal{I}_{\mathcal{A}} = (\mathcal{S}_{\mathcal{A}}, W)$ with $\mathcal{S}_{\mathcal{A}} = (S_1, \dots, S_n)$. For $1 \leq i \leq n$, define the *i-subinstance* $(\mathcal{I}_{\mathcal{A}})_i$ to be the subinstance of $\mathcal{I}_{\mathcal{A}}$ consisting of the first i squares, that is, $(\mathcal{S}_{\mathcal{A}})_i = (S_1, \dots, S_i)$. Inductively define *(unoccupied) i-pieces* to be the bounded connected maximal subspace $U_1^i, \dots, U_{n_i}^i$ of

$$([0, W] \times [0, \infty)) \setminus (\{\mathrm{BL}(S) \mid S \in (\mathcal{S}_{\mathcal{A}})_i\} \cup \{U_j^k \mid 1 \leq k < i, 1 \leq j \leq n_k\}).$$

Notice that n_i is the number of i-pieces in the packing. Often an unoccupied i-piece is just called an unoccupied piece if the value of i is redundant. An unoccupied piece is an open polygon whose boundary consists of horizontal and vertical line segments, because a piece is bounded by squares that are closed subsets and whose faces are horizontal and vertical line segments. This remark makes it possible to talk about the top, bottom, left and right face of an unoccupied piece.

Definition 4.3. Define the *top*, *bottom*, *left*, respectively *right face* of a piece U by

$$\begin{aligned} \mathrm{tf}(U) &= \max\{y \mid (x, y) \in U\}, \\ \mathrm{bf}(U) &= \min\{y \mid (x, y) \in U\}, \\ \mathrm{lf}(U) &= \min\{x \mid (x, y) \in U\}, \\ \mathrm{rf}(U) &= \max\{x \mid (x, y) \in U\}. \end{aligned}$$

Two unoccupied pieces are never adjacent, because pieces are maximal by definition. Two squares are adjacent if their intersection is non-empty. However, as squares are closed subsets and pieces are open subsets, this definition is not compatible to adjacency between a square and a piece. Hence, in the next definition, to compare a piece and a square the closure of the piece is taken. The closure of U is denoted by $\overline{U}$.

Definition 4.4. Let σ be an orthogonal packing of a SPP-instance $\mathcal{I}$. Then $S_1, S_2 \in \mathcal{S}$ are *adjacent* if $\sigma(S_1) \cap \sigma(S_2) \neq \varnothing$. And an unoccupied piece U is *adjacent* to S_1 if $\sigma(\overline{U}) \cap \sigma(S_1) \neq \varnothing$.

Next, three different kind of unoccupied pieces are defined depending on the position of the piece in the strip.

Definition 4.5. An unoccupied piece is called a *left piece* if the piece is adjacent to the left strip boundary. It is called a *right piece* if the piece is adjacent to the right strip boundary. Otherwise a piece is called a *middle piece.*

The three classes of pieces partition the collection of all pieces. Observe that a piece can never be a left piece and a right piece at the same time, because this would violate the bottom-left structure of the packing, as then there is a path contained in the piece from the left boundary to the right boundary, hence a square above the path could have been placed lower. Thus the three classes are disjoint, hence forming a partition of the collection of unoccupied pieces. There can also be unbounded unoccupied space in the packing of the strip $[0, W] \times [0, \infty)$. This unbounded unoccupied space is the topic of Section 4.5.

Next, three formal squares are defined that represent the boundary. The position of these formal squares is fixed for each packing. The main reason to introduce formal squares is to be able to generalize a lot of statements in the upcoming sections, instead of having to distinguish between left, middle and right pieces.

Definition 4.6. Let $\mathcal{I} = (\mathcal{S}, W)$ be a SSPP-instance. Define the *formal squares* $S_{\text{left}}, S_{\text{right}}$ and S_{bottom} such that under each orthogonal packing σ it holds that

$$\begin{aligned} \sigma(S_{\text{left}}) &= \{0\} \times \mathbb{R}_{\geq 0}, \\ \sigma(S_{\text{right}}) &= \{W\} \times \mathbb{R}_{\geq 0}, \\ \sigma(S_{\text{bottom}}) &= [0, W] \times \{0\}. \end{aligned}$$

Define the *formal instance of* $\mathcal{I}$ to be $\widehat{\mathcal{I}} = (\widehat{\mathcal{S}}, W)$ where $\widehat{\mathcal{S}} = \mathcal{S} \cup \{S_{\text{left}}, S_{\text{right}}, S_{\text{bottom}}\}$.

Definition 4.7. Consider a feasible packing of a (formal) SSPP-instance $\mathcal{I} = (\mathcal{S}, W)$. The *adjacency graph* $G_{\text{adj}}(\mathcal{I})$ is defined as the directed version of the graph with vertex set $\mathcal{S}$ and with an edge between two vertices if the corresponding squares are adjacent.

The adjacency graph is a connected planair digraph that describes all the adjacency relations in the packing of an instance. On the contrary, the following definition only represents the adjacency relations between (formal) squares adjacent to an unoccupied piece. This is the graph that will be used most throughout this chapter.

Definition 4.8. Let $\mathcal{I} = (\mathcal{S}, W)$ be an SSPP-instance and $\widehat{\mathcal{I}_i}$ the formal instance of $\mathcal{I}_i$. Let U be an unoccupied i-piece and assume that $\mathcal{S}' \subseteq \widehat{\mathcal{S}_i}$ are the (formal) squares adjacent to U. Then define the *adjacency graph of* U to be the adjacency graph of $\widehat{\mathcal{I}_i}$ restricted to $\mathcal{S}'$, that is,

$$G_{\text{adj}}(U) := G_{\text{adj}}(\widehat{\mathcal{I}_i})[\mathcal{S}'].$$

The adjacency graph can be seen as an undirected graph or as a digraph, because as digraph it has for every arrow another arrow going the opposite direction. The main reason to talk about arrows is to define the four different types: up, down, right and left type.

Definition 4.9. Let (S_1, S_2) be an arrow in the adjacency graph $G_{\text{adj}}(\mathcal{I})$ of a SSPP-instance.

(a) (S_1, S_2) is of *left type* if $\text{lf}(S_1) = \text{rf}(S_2)$.

(b) (S_1, S_2) is of *right type* if $\text{rf}(S_1) = \text{lf}(S_2)$.

(c) (S_1, S_2) is of *up type* if $\text{tf}(S_1) = \text{bf}(S_2)$.

(d) (S_1, S_2) is of *down type* if $\text{bf}(S_1) = \text{tf}(S_2)$.

For an illustration of the different arrow types see Figure 14. Furthermore, be cautious of the special case where two squares are only adjacent on a corner. For example, let the bottom right corner of S_1 be adjacent to the top left corner of S_2, then (S_1, S_2) is of right type as well as of down type. Moreover, notice that the arrow from the formal square S_{bottom} to S_{left} is of left type as well as of up type. Similarly, the arrow from S_{bottom} to S_{right} is of right type as well as of up type.

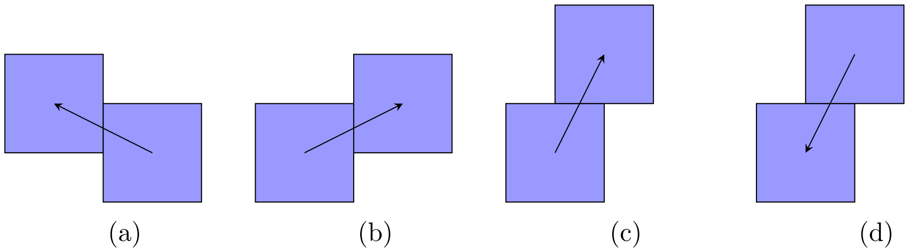


Figure 14: Left type, right type, up type and down type respectively.

4.2 The structure theorem

The structure theorem (Theorem 4.13) describes the relative position of adjacent squares in the adjacency graph of an unoccupied piece. In more detail, this section introduces a few special squares, among which are the start square and end square. There are two paths in the adjacency graph from the start square to the end square, the top path going over the unoccupied piece and the bottom path going under the unoccupied piece. Essentially, the structure theorem describes the different types of arrows that occur in the top and bottom path. In Section 4.3 and 4.4 the structure theorem will be exploited to construct a covering of the unoccupied space in a bottom-left packing.

The adjacency graph of an unoccupied piece contains a closed walk surrounding the piece, because a piece is bounded by definition and the adjacency graph contains the formal squares, hence every point on the boundary of a piece is part of a (formal) square. Actually, Lemma 4.10 will show something stronger, namely, that there exists a Hamiltonian circuit surrounding a piece in the adjacency graph of the piece. This circuit will be used to define the top and bottom paths that are required in the formulation of the structure theorem. The main idea for the proof of Lemma 4.10 is to use the natural ordering induced by the boundary of the unoccupied piece and show that this closed walk is already a Hamiltonian circuit.

Lemma 4.10. Let U be an unoccupied piece. There exists a Hamiltonian circuit in the adjacency graph of U that surrounds U.

Proof. For each square adjacent to U there is at least one point on the boundary ∂U of the piece that intersects the square and for each point on the boundary there are at most three squares intersecting this point. The boundary ∂U is homeomorphic to a circle, thus an orientation of the circle naturally gives an ordering of the squares adjacent to U. Hence this induces a closed walk W in the adjacency graph of U that surrounds U and visits each vertex at least once.

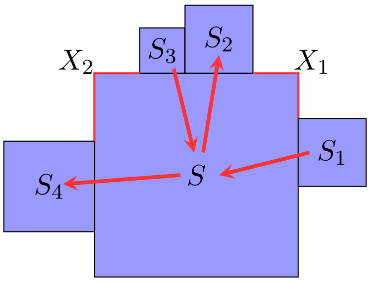


Figure 15: The arrows $(S_1, S), (S, S_2), (S_3, S)$ and (S, S_4) together with X_1 and X_2. In this example, F_1 is either the right or top face of S and F_2 is either the left or top face of S.

Claim: W is a Hamiltonian circuit, that is, W visits each square exactly once. This claim is proven by contradiction. Suppose that there is a square S that is visited twice. Then there are four arrows (S_1, S), (S, S_2), (S_3, S) and (S, S_4) such that the path from S_4 to S_1 surrounds the piece. Let $X_1 \subseteq \partial S$ be the part of the boundary of S between S_1 and S_2 that is adjacent to U. Similar, let $X_2 \subseteq \partial S$ be that between S_3 and S_4 adjacent to U. Let F_1 and F_2 be faces of S such that $F_1 \cap X_1$ respectively $F_2 \cap X_2$ do not have measure zero as subset of F_1 or F_2. Distinguish the following cases depending on the position of F_1 and F_2 relative to S.

Case 1: Let F_1 and F_2 be the same face of S. Regardless of which face F_1 is, there is a square on the path between S_2 and S_3 that can be placed more bottom-left, because exactly one of the sets X_1 or X_2 is either left or below of S_2 and S_3.

Case 2: Let F_1 and F_2 be opposite faces of S. Then X_1 and X_2 are on the same side of S_2 respectively S_3, that is, either both X_1 below S_2 and X_2 below S_3, etc. In each case, there is a square on the path between S_2 and S_3 that can be placed more bottom-left.

Case 3: Let F_1 and F_2 be two adjacent faces of S. If F_1 and F_2 are the left and bottom face of S, then the path between S_2 and S_3 can be placed more bottom-left, as this path does not surround U. Otherwise, at least one of the sets X_1 or X_2 is left or below S_2 and S_3, hence also a square on the path between S_2 and S_3 can be placed more bottom-left. □

The Hamiltonian circuit from Lemma 4.10 will be split into two directed paths, called the top path and the bottom path. These two paths start in the start square and end in the end square. The next definition defines these squares. Additionally, a few other special squares are introduced that are relevant in the subsequent sections.

Definition 4.11. Let U be an unoccupied piece and let W be the Hamiltonian circuit in the adjacency graph of U that surrounds U. Assume that W is clockwise oriented.

(a) The *start square* S_{start} is a square corresponding to a vertex in the adjacency graph of U whose top face is as low as possible and among those squares it is the left most.

(b) The *top square* S_{top} is a square corresponding to a vertex in the adjacency graph of U whose bottom face is as high as possible and among those squares it is the right most.

(c) The *end square* S_{end} is the square after S_{top} on the oriented Hamiltonian circuit W.

(d) The *pre-top square* S_{pre} is the square before S_{top} on W.

(e) The *penultimate square* S_{pen} is the square after S_{end} on W.

The adjacency graph of an unoccupied piece always has at least four vertices, because the boundary of an unoccupied piece consists of horizontal and vertical line segments, hence a rectangle is the smallest polygon enclosing unoccupied space. It follows that the start square and the top square cannot be adjacent in the adjacency graph, as then the lowest top face and the highest bottom face are on the same height, in which case the piece is empty. Consequently, the start, top, end and pre-top square are always different squares. Contrarily, the penultimate square might be the same as the start square.

The next definition splits the Hamiltonian circuit from Lemma 4.10 into two paths between the start square and the end square. This is also illustrated in Figure 16.

Definition 4.12. Let U be an unoccupied piece and let W be the Hamiltonian circuit in the adjacency graph of U that surrounds U. The *top path* is the directed path in W from the start square to the end square traversing the top square. The *bottom path* is the directed path in W from the start square to the end square traversing the penultimate square.

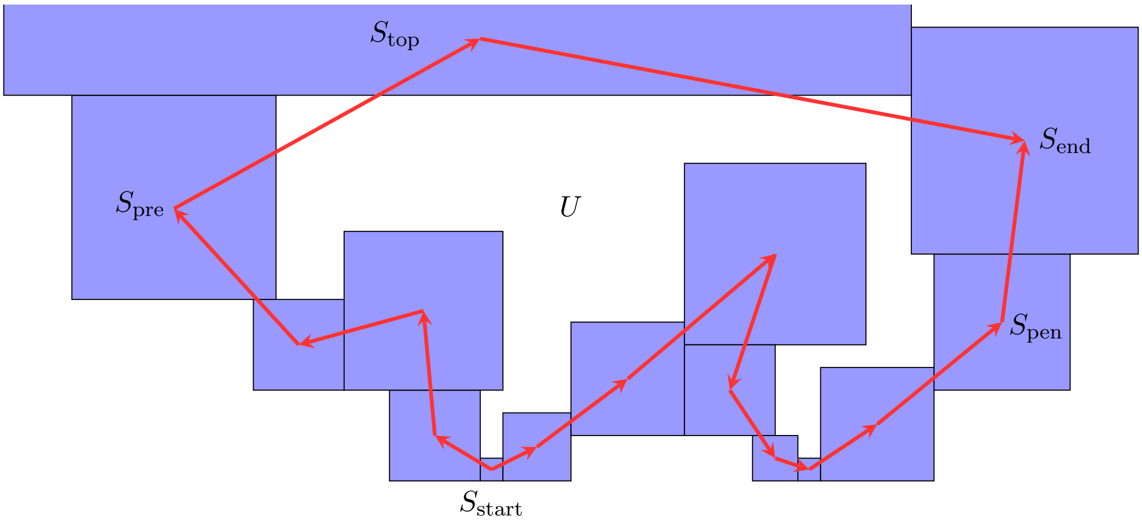


Figure 16: Example of an unoccupied piece U together with the top and bottom path in the adjacency graph. Only the squares adjacent to the piece are depicted, however, this particular example does exist in a bottom-left packing of squares. The top square is drawn as a rectangle for the sake of saving space.

Observe that the unoccupied piece is always to the right of the top path when traversed from start to end square. Analoguously, the unoccupied piece is to the left of the bottom path. Now, the structure theorem describes the types of the arrows in the top and bottom path. The structure theorem is true for left, middle and right pieces.

Theorem 4.13 (Structure theorem)**.** Let U be an unoccupied piece. Then

(a) Each arrow in the top path between the start square and the pre-top square is either of left type or of up type.

(b) The arrow from the pre-top square to the top square is either of right or up type.

(c) The arrow from the top square to the end square is either of right or down type.

(d) Each arrow in the bottom path between the start square and the penultimate square is either of right type or of down type.

(e) The arrow from the penultimate square to the end square is either of right or up type.

Proof. Let $S_0, \ldots, S_\ell$ be the top path and let $S_\ell, S_{\ell+1}, \ldots, S_{\ell+r}$ be the bottom path in reversed order. That is, $S_{\text{start}} = S_0 = S_{\ell+r}$, $S_{\text{pre}} = S_{\ell-2}$, $S_{\text{top}} = S_{\ell-1}$, $S_{\text{end}} = S_\ell$ and $S_{\text{pen}} = S_{\ell+1}$. The unoccupied piece is always on the right of the oriented circuit $S_0, \ldots, S_{\ell+r}$.

Consider the arrows (S_{start}, S_1) and $(S_{\text{start}}, S_{\ell+r-1})$. Neither (S_{start}, S_1) nor $(S_{\text{start}}, S_{\ell+r-1})$ can be of down type, as then S_1 or $S_{\ell+r-1}$ would be below S_{start}, contradicting that S_{start} is the lowest square adjacent to the unoccupied piece. Also, not both (S_{start}, S_1) and $(S_{\text{start}}, S_{\ell+r-1})$ can be of left type, as then at least one of $\text{tf}(S_1)$ and $\text{tf}(S_{\ell+r-1})$ is below $\text{tf}(S_{\text{start}})$. With the same reasoning, not both (S_{start}, S_1) and $(S_{\text{start}}, S_{\ell+r-1})$ can be of right type. Hence, as (S_{start}, S_1) is on the top path, it is of left or up type and as $(S_{\text{start}}, S_{\ell+r-1})$ is on the bottom path, it is of right or up type.

Next, consider the first arrow (S_i, S_{i+1}) on the circuit that is not of left or up type. Such an arrow exists, because if all arrows are of left and up type, then the circuit never returns to the start square. Also it holds that $i \geq 1$, because (S_0, S_1) is of left or up type by the previous paragraph. Hence it is always possible to consider the square S_{i-1}.

Case 1: Suppose (S_i, S_{i+1}) is of down type. Then (S_{i-1}, S_i) cannot be of left type, as otherwise S_i can be placed more to the left, because S_{i-1} must be above S_{i+1} and the unoccupied piece is to the right-hand side of the path S_{i-1}, S_i, S_{i+1}. Thus (S_{i-1}, S_i) must be of up type. Now as (S_i, S_{i+1}) is of down type, it follows that each arrow (S_j, S_{j+1}) with $j > i$ must be of left or up type. Otherwise, let (S_j, S_{j+1}) with $j > i$ be the first that is not of left or up type. Suppose (S_j, S_{j+1}) is of right type, then S_j could have been placed lower. And suppose (S_j, S_{j+1}) is of down type, then S_j could have been placed more to the left. Thus it follows that S_i must be the top square, i.e., $i = \ell - 1$.

Case 2: Suppose that (S_i, S_{i+1}) is of right type. Then (S_{i+1}, S_{i+2}) must be of down type as otherwise S_{i+1} could have been placed lower. Now with the same argument as before, all the arrows (S_j, S_{j+1}) with $j > i + 1$ must be of left or up type. Thus either S_i or S_{i+1} is the top square depending on which one has a bottom face that is higher. Thus this implies (a).

If $(S_{\ell-1}, S_\ell)$ is of down type, then $(S_{\ell-2}, S_{\ell-1})$ is of up type as mentioned in Case 1, this implies (b) and (c). If (S_i, S_{i+1}) is the first arrow of right type, then there are two cases. First, if $\mathrm{bf}(S_i) > \mathrm{bf}(S_{i+1})$, then S_i is the top square. It must hold that (S_{i-1}, S_i) is of up type as otherwise S_i could have been placed lower, this implies (b) and (c). Secondly, if $\mathrm{bf}(S_i) \leq \mathrm{bf}(S_{i+1})$, then S_{i+1} is the top square, this immediately implies (b), and it follows from Case 2 that also (c) is true.

All arrows on the path $S_{\ell+1}, \ldots, S_{\ell+r}$ are of left or up type according to the argumentation above. Thus each arrow in the bottom path between the start square $S_{\ell+r}$ and the penultimate square $S_{\ell+1}$ is of right or down type, this implies (d). If $(S_{\ell-1}, S_\ell)$ is of down type, then $(S_\ell, S_{\ell+1})$ is of left type, as else S_ℓ could have been placed more to the left. This implies (e). Last of all, if $(S_{\ell-1}, S_\ell)$ is of right type, then $(S_\ell, S_{\ell+1})$ must be of down type, implying (e). ☐

In line with the structure theorem, an illustration of the different types of arrows in the top and bottom path is given in Figure 17. The next section studies the consequences of the structure theorem and applies it to find a covering of the unoccupied space by squares.

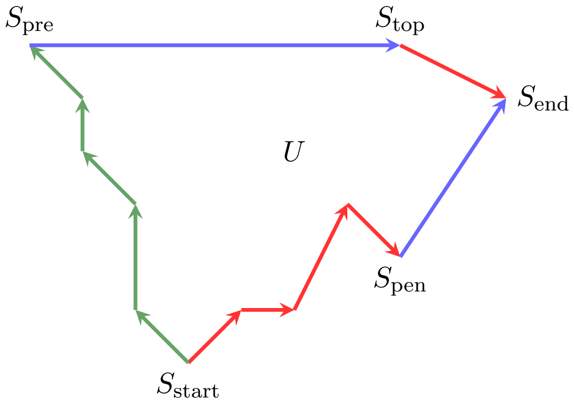


Figure 17: Example of the top and bottom path around a piece U. The green arrows are either of left or up type, the blue arrows are either of right or up type, and the red arrows are either of right or down type.

4.3 Cover partitions

The main objective of this section is to divide an unoccupied piece into easily coverable parts called subpieces. The idea of the so-called cover partition is that the width of each horizontal line in a subpiece increases with the height of the position of the horizontal line. Hence, if a square is wider than the top of a subpiece, then it is wider than any horizontal line in the subpiece. This will make it possible to cover the subpiece by the squares around it when the height of the subpiece is bounded by its width.

This section starts by defining the line space, this makes it possible to talk about horizontal lines in an unoccupied piece. After that, the cover partition is defined formally. The line space is the set of all horizontal lines in an unoccupied piece.

Definition 4.14. Let V be a subspace of an unoccupied piece U. The *(horizontal) line space of* V is defined as the set

$$\mathcal{L}V = \{\text{connected components of } \overline{V} \cap \ell_y \mid \ell_y = [0, W] \times \{y\},\ y \in [0, h_{\mathrm{BL}}]\}.$$

Here $\overline{V}$ denotes the closure of V.

To compare lines in the line space with each other, a projection map is defined. Moreover, the width of a line is defined in the obvious way.

Definition 4.15. Let V be a subspace of an unoccupied piece U. The *projection* of horizontal lines in V is the map

$$\phi \colon \mathcal{L}V \to 2^{[0,W]} : \ell = [x_0, x_1] \times \{y\} \mapsto [x_0, x_1].$$

The *width* of a horizontal line in V is defined as

$$w \colon \mathcal{L}V \to \mathbb{R}_{\geq 0} : \ell = [x_0, x_1] \times \{y\} \mapsto x_1 - x_0.$$

With these tools, the cover partition of an unoccupied piece is introduced to be the smallest partition of the piece into connected subpieces, such that a pair of projections of horizontal lines is either included in one another depending on which horizontal line is higher, or their intersection is empty.

Definition 4.16. Let U be an unoccupied piece. A *cover partition* $\mathcal{P}_{\mathrm{cov}}(U) = \{V_1, \ldots, V_s\}$ is a minimal partition of U such that

(1) For every $1 \leq i \leq s$, the subspace V_i of U is connected.

(2) For every $1 \leq i \leq s$ and every pair of horizontal lines $\ell, \ell' \in \mathcal{L}V_i$ with ℓ below ℓ', it holds that either $\phi(\ell) \subseteq \phi(\ell')$ or $\phi(\ell) \cap \phi(\ell') = \varnothing$.

Call the sets $V_1, \ldots, V_s$ the *subpieces* of U. Here minimal means that the cover partition is not the refinement of another partition satisfying (1) and (2).

Figure 19 depicts an example of a cover partition. A cover partition of an unoccupied piece is not necessarily unique. An explicit construction of a cover partition is given in Theorem 4.19. This construction will use that there is only one so-called peak in the adjacency graph of an unoccupied piece. Intuitively, a peak square is a local top square, as shown in Figure 18.

Definition 4.17. Let $S_0, \ldots, S_\ell, S_{\ell+1}, \ldots, S_{\ell+r}$ be the top path followed by the reversed bottom path in the adjacency graph of a piece U. For a vertex S_i, let S_k be the vertex with $k < i$ maximal such that $\mathrm{bf}(S_k) \neq \mathrm{bf}(S_i)$ and let S_j be the vertex with $j > i$ minimal such that $\mathrm{bf}(S_j) \neq \mathrm{bf}(S_i)$. Then S_i is a *peak square* if $\mathrm{bf}(S_k) < \mathrm{bf}(S_i)$ and $\mathrm{bf}(S_j) < \mathrm{bf}(S_i)$.

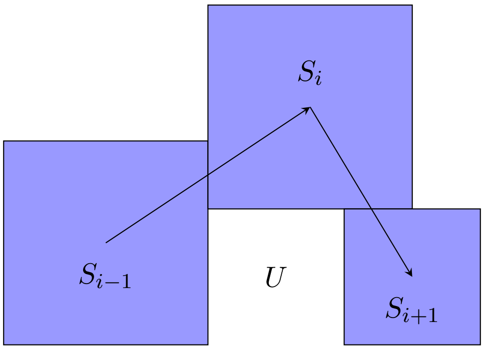


Figure 18: Example of a peak square S_i.

There can be at most two peak squares next to each other, as otherwise a peak square in the middle can be placed lower, contradicting the bottom-left placement rules. Evidently, the top square is a peak square, because it is the square adjacent to U with highest bottom face. Moreover, Lemma 4.18 shows that the structure theorem implies that there can be at most two peak squares, namely the top square, and possibly the pre-top square. This will be useful for explicitly constructing a cover partition.

Lemma 4.18. Let U be an unoccupied piece. There are at most two peak squares adjacent to U. These are the top square and possibly the pre-top square.

Proof. Let $S_0, \ldots, S_\ell, S_{\ell+1}, \ldots, S_{\ell+r}$ be the top path followed by the reversed bottom path of U, such that $S_{\ell-1}$ is the top square. Consider a peak square S_i. Let S_k with $k < i$ be the vertex closest to S_i such that $\mathrm{bf}(S_k) \neq \mathrm{bf}(S_i)$. And similarly, let S_j with $j > i$ be the vertex closest to S_i such that $\mathrm{bf}(S_j) \neq \mathrm{bf}(S_i)$. Then the bottom face of S_i is on the same height as the bottom face of all the squares S_n with $k < n < j$. Therefore, all the arrows (S_n, S_{n+1}) with $k < n < j-1$ are of right type. According to the structure theorem (Theorem 4.13), only the arrow from the pre-top square to the top square, and the arrow from the top square to the end square can be of right type. However, notice that $\mathrm{bf}(S_{\mathrm{end}}) \neq \mathrm{bf}(S_{\mathrm{top}})$, as the end square is more to the right than the top square, hence contradicting the definition of the top square. Thus for all $i \notin \{\ell-2, \ell-1\}$, with S_i a peak square, it follows that $S_k = S_{i-1}$ and $S_j = S_{i+1}$.

Now (S_{i-1}, S_i) is either of right or up type and (S_i, S_{i+1}) is either of right or down type, otherwise there is no space on the bottom face of S_i between S_{i-1} and S_{i+1} that is adjacent to U. Thus there are four combinations. First of all, let both arrows be of right type, this never happens as then S_i could have been placed lower. Secondly, let (S_{i-1}, S_i) be of right type and (S_i, S_{i+1}) be of down type, then by the structure theorem S_i is the top square or the end square. However, the end square cannot be a peak square as the bottom face of the top square is always above the bottom face of the end square, thus in this case S_i is the top square. Thirdly, let (S_{i-1}, S_i) be of up type and (S_i, S_{i+1}) be of right type, then by the structure theorem S_i is the pre-top square or the top square. Last of all, let (S_{i-1}, S_i) be of up type and (S_i, S_{i+1}) be of down type, then S_i is the top square according to the structure theorem.

All in all, the pre-top and the top are the only possible peak squares. Moreover, if both are peak squares, then the bottom faces of the peak squares are on the same height because the pre-top square and the top square are adjacent. □

Finally, Theorem 4.19 gives an explicit construction of a cover partition, called the natural cover partition. An example of a natural cover partition is given in Figure 19. In Section 4.4 this construction is used to inductively cover an unoccupied piece.

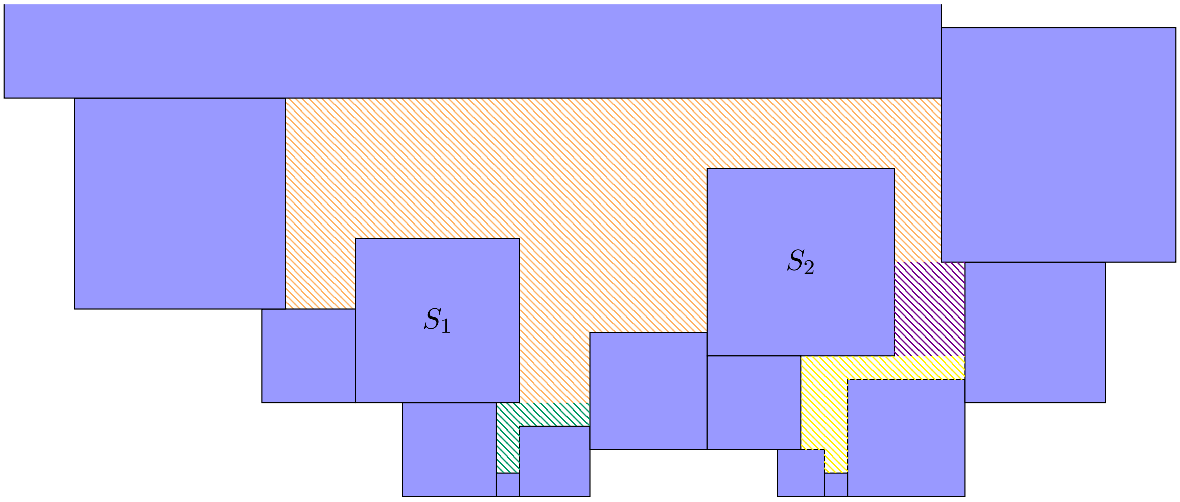


Figure 19: Example of the natural cover partition of an unoccupied piece. There are four subpieces indicated by the green, yellow, purple and orange area. Green is V_1, yellow is V_2, purple is V_{end} and orange is V_{top}. The top square is drawn as a rectangle for the sake of convenience.

Theorem 4.19. Let U be an unoccupied piece. There exists an explicit construction of a cover partition of U. Call this the *natural cover partition.*

Proof. Let $S_0, \dots, S_\ell$ be the top path of U and let $S_\ell, \dots, S_{\ell+r}$ be the bottom path in reversed order. Consider the path $P = S_\ell, \dots, S_{\ell+r} = S_0, \dots, S_{\ell-1}$. Let S_{i_j} be the j-th vertex on P such that $\mathrm{rf}(S_{i_j}) > \mathrm{rf}(S_{i_j-1})$ and (S_{i_j-1}, S_{i_j}) is of up type. This gives a sequence of squares $S_{i_1}, \dots, S_{i_s}$ on the path P. Furthermore, the structure theorem (Theorem 4.13) states that there is at most one arrow on path P of down type, this is the arrow from the end square to the penultimate square, that is, $(S_\ell, S_{\ell+1})$.

If $(S_\ell, S_{\ell+1})$ is of down type, then consider the sequence of vertices $S_\ell, S_{i_1}, \dots, S_{i_s}$. Otherwise just consider the sequence $S_{i_1}, \dots, S_{i_s}$ without the end square S_ℓ. Sort $\mathrm{bf}(S_\ell), \mathrm{bf}(S_{i_1}), \dots, \mathrm{bf}(S_{i_s})$ by increasing height and partition U in this order. Inductively define the subpiece V_j to be the space below $\mathrm{bf}(S_{i_j})$ that is connected to the bottom face of S_{i_j} and disjoint from the already constructed subsets of U. For the end square S_ℓ denote the subpiece by V_{end}. At last, let V_{top} be the space connected to the bottom face of the top square and disjoint from $V_{\text{end}}, V_1, \dots, V_s$. In the end this gives a collection of subspaces $\mathcal{P}_{\text{cov}} = \{V_1, \dots, V_s, V_{\text{end}}, V_{\text{top}}\}$ (or without V_{end}). Discard the sets of measure zero. Call S_{i_j} the *square corresponding to* V_j.

Claim 1: The collection $\mathcal{P}_{\text{cov}}$ partitions U. To prove this claim, observe that by definition the sets are disjoint. Furthermore, suppose that there is a point p in U that is not covered by the sets in $\mathcal{P}_{\text{cov}}$. Then also none of the space directly above p is covered by the sets, nor any space next to this vertical line. In other words, this uncovered space must be adjacent to a peak square. By Lemma 4.18 the top square and possibly the pre-top square are the only peak squares. Hence the point p is actually covered by V_{top}. Thus, it follows that the sets in $\mathcal{P}_{\text{cov}}$ cover all the space in U. Hence these sets form a partition of U.

Claim 2: The collection $\mathcal{P}_{\text{cov}}$ forms a cover partition of U. In other words, it has to be shown that the sets in $\mathcal{P}_{\text{cov}}$ are a minimal partition satisfying (1) and (2) from Definition 4.16. Property (1), connectedness of the sets in $\mathcal{P}_{\text{cov}}$, follows by definition. For property (2) use contradiction. Suppose that $V \in \mathcal{P}_{\text{cov}}$ does not satisfy property (2), then there exists a pair of lines $\ell, \ell' \in \mathcal{L}V$ with ℓ below ℓ' such that $\phi(\ell) \cap \phi(\ell') \neq \varnothing$ and one of the endpoints of $\phi(\ell)$ is outside of $\phi(\ell')$. Now distinguish two cases.

Case 1: The left endpoint of $\phi(\ell)$ is outside of $\phi(\ell')$. Let S be a square on the path P intersecting the left endpoint of ℓ and let S' be a square on the path P intersecting the left endpoint of ℓ' Then it holds that $\mathrm{rf}(S') > \mathrm{rf}(S)$ and S lies before S' on the path P. It follows that there is a vertex S_{i_m} in the constructed sequence that is between S and S' on the path P. Therefore, the spaces V_m corresponding to S_{i_m} and V overlap, contradicting that $\mathcal{P}_{\mathrm{cov}}$ is a partition.

Case 2: The right endpoint of $\phi(\ell)$ is outside of $\phi(\ell')$. Now, let S be a square in the path P intersecting the right endpoint of ℓ and S' be a square in the path P intersecting the right endpoint of ℓ'. It holds that $\mathrm{lf}(S') < \mathrm{lf}(S)$ and S' comes before S on the path P. These two properties can only be satisfied if there is an arrow of right or down type between S' and S on the path P. By the structure theorem (Theorem 4.13), there does not exists a right type arrow on the path P. As mentioned above, the only arrow on the path P of down type is the arrow from the end square to the penultimate square. Now it follows that V_{end} intersects V. This contradicts that $\mathcal{P}_{\mathrm{cov}}$ forms a partition.

Finally, it remains to show that the partition is minimal. Let V_j and $V_{j'}$ be two sets in the partition with $j < j'$. Suppose that the union is connected, then $V_{j'}$ is above V_j, as otherwise V_j could have been chosen larger in the inductive definition. Now the square corresponding to V_j shows that $V_j \cup V_{j'}$ cannot satisfy property (2). Thus $V_j \cup V_{j'}$ is either disconnected or does not satisfy property (2), thus the cover partition is minimal. This proves Claim 2. The constructed cover partition is called the natural cover partition. □

In the following, it will be useful to switch between a subpiece V_j of the natural cover partition and the square S_{i_j} corresponding to this subpiece in the inductive definition of the proof of Theorem 4.19.

Definition 4.20. Let $\mathcal{P}_{\mathrm{cov}}(U) = \{V_1, \dots, V_s, V_{\mathrm{end}}, V_{\mathrm{top}}\}$ be the natural cover partition of an unoccupied piece. The *square corresponding to V_j* is the square S_{i_j} in the path P such that $\mathrm{rf}(S_{i_j}) > \mathrm{rf}(S_{i_j-1})$ and (S_{i_j-1}, S_{i_j}) is of up type. The end square is the *square corresponding to V_{end}* and the top square is the *square corresponding to V_{top}*.

The square corresponding to the green area in Figure 19 is S_1 and the square corresponding to the yellow area is S_2. Furthermore, the end square is the square corresponding to the purple area and the top square is the square corresponding to the orange area.

4.4 The local cover theorem

The preceding sections studied the relation of squares relative to each other in the bottom-left placement of a SSPP-instance. However, it was never actually used that all rectangles were squares, that is, the statements would also hold for SPP-instances. On the contrary, this section will make extensive use of the fact that all rectangles are squares. The fact that the width and height of a square are equal is important information to find a feasible covering of the unoccupied space.

This section starts by defining the width and height of an unoccupied piece and its subspaces. Next, Lemma 4.23 relates the size of a subpiece in the natural cover partition to the size of the square corresponding to the subpiece. This will be the bedrock of the local cover theorem (Theorem 4.24) that covers a left or middle unoccupied piece by at most four copies of the squares that are adjacent to it. Furthermore, Theorem 4.26 shows that right pieces can also be covered by at most four copies of the squares adjacent to it, however, this requires more work as part of the right boundary of a right piece is the right strip boundary. Section 4.5 will discuss how to cover the remaining unoccupied space and Section 4.6 unites all these local coverings into a global covering that uses at most twelve copies of the squares of the instance to cover all the unoccupied space.

The width of a subspace of an unoccupied piece is defined as the maximum width of a line in its linespace. The height of an unoccupied piece is defined as the height of its bounding box, that is, of the smallest rectangle bounding the piece.

Definition 4.21. Let V be a subspace of an unoccupied piece U. The *width* of V is defined as

$$w(V) = \max\{w(\ell) \mid \ell \in \mathcal{L}V\}.$$

The *height* of V is defined as

$$h(V) = \sup\{y - y' \mid [x_0, x_1] \times \{y\}, [x_0', x_1'] \times \{y'\} \in \mathcal{L}V\}.$$

For a cover partition, the fact that higher lines in a subpiece have larger width implies that the width of a subpiece is always attained in the highest line.

For each square in the bottom-left packing there is a (formal) square adjacent to the left face as otherwise the square could have been placed more to the left. Similarly, there is a (formal) square adjacent to the bottom face, as otherwise the square could have been placed lower. Inductively it follows that for each square there is a path in the adjacency graph consisting of right type arrows from the left strip boundary to the square. Analoguously, there is a path consisting of up arrows from strip bottom to the square. The next definition describes two subgraphs of the adjacency graph of an instance that contain such paths.

Definition 4.22. The *right type adjacency graph* $G_{\text{right}}(\mathcal{I})$ is the subgraph of the adjacency graph $G_{\text{adj}}(\mathcal{I})$ consisting of all vertices together with all right type arrows. Similarly, the *up type adjacency graph* $G_{\text{up}}(\mathcal{I})$ is the subgraph consisting of all up type arrows.

Lemma 4.23 relates the width of a subpiece in the natural cover partition to the width of the square corresponding to the subpiece. Moreover, in the case that the arrow in the bottom path from the penultimate square to the end square is of up type, the lemma says even more about the width of the pretop square and top square. In the proof it will be significant that an i-piece is not yet bounded when only the first $i-1$ squares of the instance are placed.

Lemma 4.23. Let U be an i-piece. Let $\mathcal{P}_{\text{cov}}(U) = \{V_1, \dots, V_s, V_{\text{end}}, V_{\text{top}}\}$ be the natural cover partition of U for the subinstance $\mathcal{I}_i$ consisting of the first i squares of SSPP-instance $\mathcal{I}$. Then

(a) For every square S corresponding to subpiece $V \in \mathcal{P}_{\text{cov}}(U)\backslash\{V_{\text{end}}\}$, it holds that $S > w(V)$.

Furthermore, if the arrow from the penultimate square to the end square is of up type, then

(b) The pretop square is larger than the height difference between the bottom face of the top square and the bottom face of the end square, that is, $S_{\text{pre}} > \text{bf}(S_{\text{top}}) - \text{bf}(S_{\text{end}})$.

(c) The top square is larger than the penultimate square plus the width of a line in the line space of U just under the bottom face of the end square, that is, $S_{\text{top}} > S_{\text{pen}} + w(V_{\text{end}})$.

Proof. Let S be the square corresponding to $V \in \mathcal{P}_{\text{cov}}(U) \setminus \{V_{\text{end}}\}$ and let $\ell = [x, x+w(V)] \times \{y\}$ be the highest line in $\mathcal{L}V$. Consider the space $K = [x, x + w(V)] \times [y, y + S]$ of width $w(V)$ and height S above the line ℓ. Notice that $S \cap K$ is non-empty, as part of the line ℓ is adjacent to the bottom face of S. Assume that S is the j-th square that is placed.

Claim 1: The interior of K is unoccupied when only the first $j-1$ squares are placed. This claim is proven by contradiction. Suppose that there is a square S' intersecting the interior of K. Then there is a path P_{right} from the left strip boundary to S' in the right type adjacency graph $G_{\text{right}}(\mathcal{I}_{j-1})$ consisting of the first $j-1$ squares. Similarly, there is a path P_{up} from the strip bottom to S' in $G_{\text{up}}(\mathcal{I}_{j-1})$. It holds that $\text{bf}(S') < \text{tf}(S)$ and $\text{lf}(S') > \text{rf}(S)$, because S' intersects the interior of K. It follows that the path P_{up} is to the right of S. Furthermore, suppose that P_{right} goes over S, then U was already bounded by the paths P_{right} and P_{up} when the first $j-1$ squares were placed, contradicting the definition of an i-piece. Thus, the path P_{right} goes underneath S. This means that P_{right} crosses V, because V is just under K and all arrows in P_{right} are of right type, hence this contradicts that V is unoccupied. This proves the claim.

Suppose that $S \leq w(V)$, then S could have been placed lower as K is unoccupied. Therefore, it holds that $S > w(V)$. This proves part (a).

Now, assume that the arrow from the penultimate square to the end square is of up type. **Claim 2**: The end square is the i-th square of the instance, that is, it is the square that bounds U. The path from the end square to the left strip boundary in the right type adjacency graph has to go over U, because the arrow from the penultimate square to the end square is of up type and the unoccupied piece is on the left of the bottom path. Thus when the end square is placed the piece is bounded, implying the claim.

The structure theorem (Theorem 4.13) implies that the arrow from the pretop square to the top square is of up type and the arrow from the top square to the end square is of right type Now, suppose that $S_{\text{pre}} \leq \text{bf}(S_{\text{top}}) - \text{bf}(S_{\text{end}})$, then as the pretop square is placed before the end square, and $\text{bf}(S_{\text{top}}) - \text{bf}(S_{\text{end}}) \leq S_{\text{end}}$, it follows that the pretop square could have been placed lower at the position of the end square. This proves part (b).

Finally, part (b) implies that $\text{bf}(S_{\text{end}}) \geq \text{bf}(S)$ for all $S \in \mathcal{P}_{\text{cov}} \setminus \{S_{\text{top}}\}$. Similar to part (a), consider the highest horizontal line $\ell = [x, x + w(V_{\text{end}})] \times \{y\}$ in the line space $\mathcal{L}V_{\text{end}}$ and define the space $K = [x, x + w(V_{\text{end}}) + S_{\text{pen}}] \times [y, y + S_{\text{top}}]$ of width $w(V_{\text{end}}) + S_{\text{pen}}$ and height S_{top} above the line ℓ. Now, the interior of K is unoccupied when only the squares before the top square are placed. This is proven with contradiction in the same way as Claim 1. Next, suppose that it holds that $w(S_{\text{top}}) \leq w(\ell) + w(S_{\text{pen}})$, then the top square could have been placed lower as K is unoccupied. Therefore, it holds that $w(S_{\text{top}}) > w(\ell) + w(S_{\text{pen}})$. This implies part (c). □

Finally, the local cover theorem (Theorem 4.24) constructs a cover of a left or middle unoccupied i-piece using at most four copies of the squares adjacent to the piece. The main idea is to cover each subpiece in the natural cover partition by at most two copies of the square S corresponding to the subpiece, this is possible as the width of the subpiece is smaller than S. However, the height of the subpiece might be larger than S, in that case, the other square adjacent to the subpiece on the same height as S is also used to cover the subpiece.

Theorem 4.24 (Local cover theorem)**.** *Let U be a left or middle i-piece in the bottom-left packing of the subinstance $\mathcal{I}_i$. Then U can be covered by at most four copies of the squares that are adjacent to U.*

Proof. Let $\mathcal{P}_{\text{cov}}(U) = \{V_1, \ldots, V_s, V_{\text{end}}, V_{\text{top}}\}$ be the natural cover partition of U. Distinguish two cases depending on the type of the arrow from the penultimate square to the end square.

Case 1: Let the arrow from the penultimate square to the end square be of right type. Then let $S_{i_1}, \ldots, S_{i_s}$ be the squares corresponding to $V_1, \ldots, V_s$ and let S_{top} be the top square corresponding to V_{top}. Observe that V_{end} does not exist. For $1 \leq j \leq s$, let Q_j be the path from S_{i_j-1} to S_{end} that traverses the top path in reversed order followed by the bottom path. And define S_{k_j} to be the first vertex on the path Q_j such that $\text{tf}(S_{k_j}) > \text{bf}(S_{i_j})$. Obviously, S_{k_j} is adjacent to V_j as otherwise V_j could have been chosen to be larger. Now split V_j into $V_j^{\perp}$ and $V_j^{\top}$, where $V_j^{\perp}$ is everything below the bottom face of S_{k_j} and $V_j^{\top}$ everything above the bottom face of S_{k_j}, that is,

$$V_j^{\perp} = V_j \cap ([0, W] \times [0, \mathrm{bf}(S_{k_j})]) \quad \text{and} \quad V_j^{\top} = V_j \cap ([0, W] \times [\mathrm{bf}(S_{k_j}), \infty)).$$

The structure theorem (Theorem 4.13) states that all arrows in the path Q_j are either of right or down type. Thus for each square on Q_j, the effect it has on the height of V_j is less than the effect it has on the width of V_j. Therefore it follows that $w(V_j^{\perp}) > h(V_j^{\top})$. Now by part (a) of Lemma 4.23 it holds that $S_{i_j} > w(V_j)$, hence S_{i_j} can cover $V_j^{\top}$. Furthermore, if $S_{i_j} \geq S_{k_j}$, then S_{i_j} can also cover $V_j^{\top}$. Otherwise if $S_{i_j} < S_{k_j}$, then S_{k_j} can cover $V_j^{\top}$. This is illustrated in Figure 20.

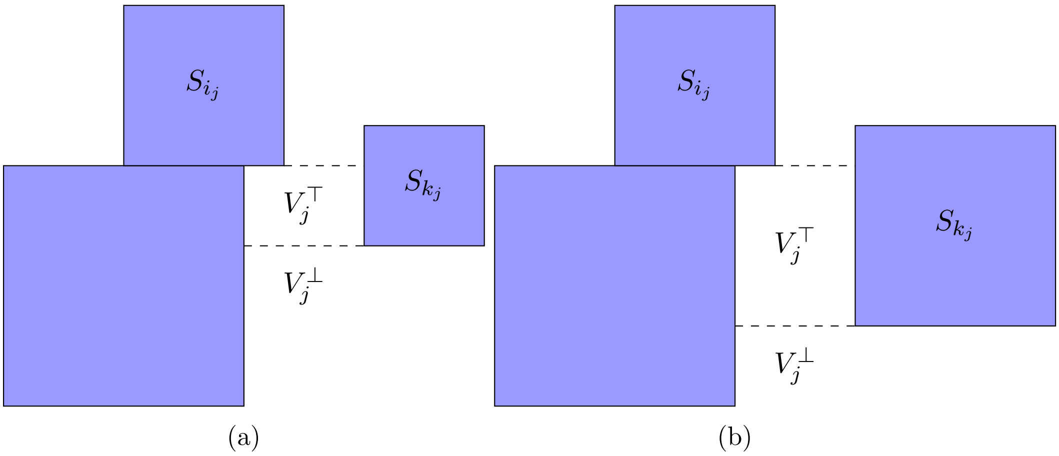


Figure 20: In Figure 20a the square corresponding to S_{i_j} is larger than the square corresponding to S_{k_j}, so S_{i_j} is used twice to cover $V_j^{\top}$ and $V_j^{\perp}$. Furthermore, in Figure 20b the square corresponding to S_{i_j} is smaller than the square corresponding to S_{k_j}, so S_{i_j} is used once to cover $V_j^{\perp}$ and the area in S_{k_j} directly next to $V_j^{\top}$ is used once to cover $V_j^{\top}$.

In the same way, let $V_{\text{top}}^{\perp}$ be all the space in V_{top} under $\mathrm{bf}(S_{\text{end}})$ and $V_{\text{top}}^{\top}$ all the space in V_{top} above $\mathrm{bf}(S_{\text{end}})$. Again, part (a) of Lemma 4.23 states that $S_{\text{top}} > w(V_{\text{top}})$, thus either two copies of S_{top} or one copy of S_{top} and one copy of S_{end} cover V_{top}.

All in all, at most three copies of the squares adjacent to U cover the unoccupied piece. Namely, for each subpiece use at most two copies of S_{i_j} and at most one copy of S_{k_j}. Observe that if S_{k_j} is used to cover $V_j^{\top}$, then only $S_{k_j} \cap ([0, W] \times [\mathrm{bf}(V_j^{\top}), \mathrm{tf}(V_j^{\perp})])$ is needed, thus S_{k_j} can be used to cover multiple subpieces. Thus each square might be used at most three times in the covering of the piece U.

Case 2: Let the arrow from the penultimate square to the end square be of up type. Now, let $S_{i_1}, \ldots, S_{i_s}$ be the squares corresponding to $V_1, \ldots, V_s$, let S_{top} the square corresponding to V_{top} and S_{end} the square corresponding to V_{end}. Cover V_{top} and V_j for $1 \leq j \leq s$ in the same way as Case 1.

Next, the subpiece V_{end} is covered by at most two copies of the top square. Again split V_{end} into $V_{\text{end}}^{\perp}$ and $V_{\text{end}}^{\top}$, where $V_{\text{end}}^{\perp}$ is everything in V_{end} that is under $\mathrm{bf}(S_{\text{pen}})$, and $V_{\text{end}}^{\top}$ everything above $\mathrm{bf}(S_{\text{pen}})$. By part (c) of Lemma 4.23 it holds that $S_{\text{top}} > S_{\text{pen}} + w(V_{\text{end}})$, this implies that $S_{\text{top}} > S_{\text{pen}}$ and $S_{\text{top}} > w(V_{\text{end}})$. Now by the same argumentation as before, the structure theorem (Theorem 4.13) states that all arrows on the bottom path are of right or up type, therefore, each square in the bottom path adjacent to $V_{\text{end}}^{\perp}$ adds more to the width than to the height, thus $w(V_{\text{end}}^{\perp}) > h(V_{\text{end}}^{\perp})$. This implies that the top square can cover $V_{\text{end}}^{\perp}$. Furthermore, $S_{\text{top}} > S_{\text{pen}}$ implies that the top square can cover $V_{\text{end}}^{\top}$.

All in all, at most four copies of the squares adjacent to U cover the unoccupied piece. Each subpiece V_j uses at most two copies of S_{i_j}. At most one copy is needed of a square that plays the role of some S_{k_j}. And the top square is used at most four times, namely, at most twice to cover V_{top} and at most twice to cover V_{end}. As a top square is never used in the role of S_{k_j}, it follows that at most four copies of the squares adjacent to U suffices to cover the piece. □

Later, the global cover theorem (Theorem 4.32) will construct a global covering of the unoccupied space using the local cover theorem (Theorem 4.24) multiple times. It will be crucial that each square is only used a limited number of times to cover different local coverings. As mentioned in the end of the previous proof, a square can be used to cover unoccupied space in two different roles. First of all, a square can be used to cover a subpiece V_j in the role of the square S_{i_j} corresponding to V_j. Secondly, a square can be used to cover $V_j^{\top}$ in the role of the square S_{k_j}, which is the first square on the path Q_j such that the top face of S_{k_j} is strictly above the bottom face of S_{i_j}. The next definition makes this distinction formal.

Definition 4.25. A square S is a *left cover square* if S is used to cover a subpiece V_j of an unoccupied piece U in the role of the square S_{i_j} corresponding to V_j. A square is a *right cover square* if S is used to cover a subpiece V_j in the role of the square S_{k_j}, which is the first square on the path Q_j between the square S_{i_j-1} and the end square containing the start square such that $\text{tf}(S_{k_j}) > \text{bf}(S_{i_j})$.

The local cover theorem (Theorem 4.24) gives a recipe for covering left and middle pieces, thus it remains to construct a covering for right pieces. The covering from before does not work, because for a subpiece V_j the right cover square S_{k_j} on the path Q_j might be the formal right strip boundary square and this cannot be used to cover V_j. Despite of this, Theorem 4.26 constructs a cover that heavily relies on the local cover theorem (Theorem 4.24). Namely, it uses the local cover theorem for subpieces that have a right cover square adjacent to it, and it uses other squares in the top path to cover the other unoccupied space.

Theorem 4.26 (Right piece local cover theorem)**.** *Let U be a right i-piece in the bottom-left packing of the subinstance $\mathcal{I}_i$. Then U can be covered by at most four copies of the squares adjacent to U.*

Proof. Let $\mathcal{P}_{\text{cov}}(U) = \{V_1, \dots, V_s, V_{\text{top}}\}$ be the natural cover partition of U. Note that V_{end} does not exist for a right piece. For each subpiece V_j such that S_{k_j} does not equal the formal right strip boundary, use the same covering with S_{i_j} and S_{k_j} as in Theorem 4.24. Now let $V_{l_1}, \dots, V_{l_t}$ be the other subpieces that are not yet covered (including V_{top}).

For each $1 \leq j \leq t$ it holds that $S_{i_{l_j}} > w(V_{l_j})$ by part (a) of Lemma 4.23. However, the height of V_{l_j} can be significantly larger than $S_{i_{l_j}}$, because there is not a bottom path consisting of right and down type arrows on the right of the subpiece. Instead, let ℓ be the bottom-left most horizontal line in $\mathcal{L}V_{i_l}$ and let S be a square adjacent to the left endpoint of ℓ. Consider the path $w_1, \dots, w_q$ from S to $S_{i_{l_j}}$. By the structure theorem (Theorem 4.13) this path consists of left and up type arrows. Next the piece V_j is partitioned into q classes $W_1, \dots, W_q$ where W_k is the unoccupied space in the rectangle $[\text{rf}(w_k), W] \times [\text{tf}(w_{k-1}), \text{tf}(w_k)]$ that is connected to the square w_k. These sets form a partition, because if there is other unoccupied space in the rectangle corresponding to W_k, then there is another square $w_{k'}$ on the top path such that right of that square there is another set $W_{k'}$ of the partition. Each W_k is covered by either w_k, w_{k+1} or S_{i_j} depending on the arrow types of (w_{k-1}, w_k) and (w_k, w_{k-1}).

Case 1: Let (w_{k-1}, w_k) and (w_k, w_{k-1}) be of left type. Then the effect of w_k on the height of V_j is strictly less than the effect on the width of V_j. Thus cover W_k with S_{i_j}.

Case 2: Let (w_{k-1}, w_k) and (w_k, w_{k-1}) be of up type. Then it holds that $w(W_k) < w_{k+1}$, because otherwise w_{k+1} could have been placed lower. Now if $w_k < w_{k+1}$, then as $w_k = h(W_k)$ it is possible to cover W_k with w_{k+1}. Otherwise if $w_k \geq w_{k+1}$, then it holds that $w_k > w(W_k)$, thus W_k can be covered with w_k.

Case 3: Let (w_{k-1}, w_k) be of left type and (w_k, w_{k-1}) of up type. This is similar to Case 2. If $w_k < w_{k+1}$, then cover W_k with w_{k+1}. Otherwise cover W_k with w_k.

Case 4: Let (w_{k-1}, w_k) be of up type and (w_k, w_{k-1}) of left type. Then the effect of w_k on the height of V_j equals the effect on the width of V_j. Thus cover W_k with S_{i_j}.

Let $W_{m_1}, \ldots, W_{m_r}$ be the sets from the partition that were covered with S_{i_j} in Case 1 and 4. As mentioned above, for each W_k it holds that $S_{i_j} > w(W_k)$. Furthermore, as the effect of w_{m_k} on the height of V_j is less or equal to the effect on the width of V_j it follows that $\sum_{x=1}^{r} w_{m_x} \geq \sum_{x=1}^{r} h(W_{m_x})$. Hence it holds that

$$S_{i_j} > w(V_j) \geq \sum_{x=1}^{r} w_{m_x} > \sum_{x=1}^{r} h(W_{m_x}).$$

Thus one copy of S_{i_j} is enough to cover $W_{m_1}, \ldots, W_{m_r}$.

All in all, a square is used at most three times to cover subpieces of the natural cover partition by the local cover theorem (Theorem 4.24), because the arrow from the penultimate square to the end square is of right type. Furthermore, a square can be in at most two of the four cases above. Hence at most two copies of a square are needed to cover the $W_1, \ldots, W_q$, namely, at most once in Case 2, at most once in Case 3 and at most once in Case 1 and 4. Moreover, a square that is used more than twice to cover subpieces in the local cover theorem can only belong to Case 1 above (or to none of the cases). This is true because such a square is a left and right cover square, hence there is unoccupied space in U adjacent to the left face and to the right face of the square. Therefore, it follows that the arrow to and from this square in the top path must be of left type by the construction in Lemma 4.10. Therefore it follows that in total at most four copies of the squares adjacent to the right piece are needed to cover it. □

4.5 Trenches

The unoccupied space of a bottom-left packing is usually larger than just the unoccupied pieces. In particular, there can be space in the top of the packing that is unbounded in the strip. These subspaces are called trenches. Informally, a trench looks like an unoccupied piece without a top square. Theorem 4.30 and Theorem 4.31 reduce trenches to left, middle or right pieces to show that part of a trench can be covered by at most four copies of squares that are adjacent to it. This will be used in Section 4.6 to construct a global covering of all the unoccupied space of the packing.

Definition 4.27. Let $\mathcal{I}$ be a SSPP-instance and let $\mathcal{U}$ be the set of all unoccupied pieces. A *trench* is a bounded connected maximal subspace T of

$$([0, W] \times [0, h_{\mathrm{BL}} - h_{\max}]) \setminus (\mathrm{BL}(\mathcal{I}) \cup \mathcal{U}).$$

Let $\mathcal{T} = \{T_1, \ldots, T_s\}$ be the set of trenches.

All trenches have the same properties except for one special trench called the right trench. The right trench is the trench that is adjacent to the right strip boundary. In the reduction theorems, the right trench is reduced to a right piece instead of a left or middle piece.

Definition 4.28. The *right trench* is the unique trench that is adjacent to the right strip boundary. The other trenches are called *top trenches.*

The definition of trenches directly implies the following corollary describing the unoccupied space in a bottom-left packing in terms of unoccupied pieces and trenches.

Corollary 4.29. The unoccupied space of a bottom-left packing in the $[0, W] \times [0, h_{\mathrm{BL}} - h_{\max}]$ substrip equals $\mathcal{U} \cup \mathcal{T}$, the union of the unoccupied pieces and the trenches.

Next, Theorem 4.30 shows that top trenches can be covered by at most four copies of the squares adjacent to it when a substrip of height $3h_{\max}$ is cut off the top of the strip. Section 4.7 will show that cutting of such a substrip does not effect the boundedness of the approximation ratio. The main idea for covering the top trenches is to first reduce the top trenches to left and middle pieces, and then use the local cover theorem (Theorem 4.24) to cover the unoccupied space with squares adjacent to the trench.

Theorem 4.30 (Top trench reduction theorem)**.** Let T be a top trench. Then the space

$$T \cap ([0, W] \times [0, h_{\mathrm{BL}} - 3h_{\max}])$$

can be covered by at most four copies of the squares adjacent to T.

Proof. Let ℓ be the highest horizontal line in the line space of T. There is a square above the line ℓ, since at least one square touches the top of the strip at height h_{BL}, hence it holds that $w(\ell) < h_{\max}$. Place a square S of size $h_{\max}$ on top of the line ℓ to reduce the top trench T to a left or middle piece. Use the local cover theorem (Theorem 4.24) to cover the piece with at most four copies of the squares adjacent to T. The square S of size $h_{\max}$ is the top square of the unoccupied piece T. Now S is used at most four times to cover T, and S is only used to cover unoccupied space that is at most $2h_{\max}$ under the line ℓ. Therefore, all the space in $T \cap ([0, W] \times [0, h_{\mathrm{BL}} - 3h_{\max}])$ is covered by actual squares that are adjacent to T. $\square$

It remains to cover the right trench. This is done similar to top trenches. However, the difference is that the right trench is reduced to a right piece by adding a copy of the largest square of the instance to the top of the trench. Next, Theorem 4.26 shows a way to cover the right piece by at most four copies of the squares adjacent to it.

Theorem 4.31 (Right trench reduction theorem)**.** The right trench T can be covered by at most four copies of the squares adjacent to T together with a square of size $h_{\max}$.

Proof. Let ℓ be the highest horizontal line in the line space of T. There is a square above the line ℓ, hence $w(\ell) < h_{\max}$. Place a square S of size $h_{\max}$ on top of the line ℓ to reduce the right trench T to a right piece. Use Theorem 4.26 to cover T by at most four copies of the squares adjacent to the right piece T. It follows that this space can be covered by at most four copies of the squares that are actual adjacent to the right trench T together with a square of size $h_{\max}$. $\square$

4.6 The global cover theorem

The global cover theorem (Theorem 4.32) combines the different reductions with the local cover theorems to obtain a global covering of the unoccupied space in a bottom-left packing of squares restricted to the $[0, W] \times [0, h_{\mathrm{BL}} - 3h_{\max}]$ substrip. The main caveat is to show that each square is only used a limited number of times to cover unoccupied space locally. Essentially, the reductions from the trenches to the unoccupied pieces show that each square covers unoccupied space as left cover square, as right cover square, or it covers a right piece. The idea of the global cover theorem is to show that if a square is used multiple times as left or right cover square, then still only a limited number of copies of this square is required.

Theorem 4.32 (Global cover theorem)**.** Let $\mathcal{I}$ be a SSPP-instance. The unoccupied space in the $[0, W] \times [0, h_{\mathrm{BL}} - 3h_{\max}]$ substrip can be covered by at most twelve copies of the squares from the instance.

Proof. By Corollary 4.29 all the unoccupied space of the substrip $[0, W] \times [0, h_{\mathrm{BL}} - 3h_{\max}]$ is contained in the union of the pieces and trenches. Use the local cover theorem (Theorem 4.24) to cover the left and middle unoccupied pieces, use Theorem 4.26 to cover the right pieces, use Theorem 4.30 to cover the top trenches and use Theorem 4.26 to cover the right trench. It remains to show that each square is only used a limited number of times to cover unoccupied space.

Claim 1: If a square S is used multiple times as left cover square, then still at most six copies of the square are needed. Let $U_1, \ldots, U_l$ be the different pieces that use S as left cover square in their local covering. Then S is the top square of each U_j except possibly for one such piece. Without loss of generality, let S be the top square of the pieces $U_1, \ldots, U_{l-1}$. Let $V_1, \ldots, V_{l-1}$ be the subpieces of the natural cover partition of $U_1, \ldots, U_{l-1}$ that use S in their local covering. Obviously, it holds that $\sum_{j=1}^{l-1} w(V_j) < S$ as S is the top square of all these pieces. Furthermore, the top faces of the end squares of $U_1, \ldots, U_{l-1}$ are on the same height and the arrow from the top square S of U_j to the end square of U_j is of down type. This implies that the arrow from the penultimate square to the end square of U_j for $1 \leq j \leq l-1$ is of right type. Hence, to cover V_j for $1 \leq j \leq l-1$ at most two copies of the space in S directly above V_j are needed. Furthermore, at most four copies of S are required to cover V_l according to the local cover theorem (Theorem 4.24). Thus in total at most six copies of S are needed for S in the role of left cover square to cover the unoccupied space.

Claim 2: If a square is used multiple times as right cover square, then at most one copy of the square is needed. To prove this claim, let $V_1, \ldots, V_l$ be the different subpieces that use S as right cover square in their local covering. These subpieces might belong to different pieces. For each j, the square S is used to cover $V_j^{\top}$ as defined in the proof of the local cover theorem (Theorem 4.24). Now it holds that $\sum_{j=1}^{l} h(V_j^{\top}) < S$, because each point on the right boundary of $V_j^{\top}$ is adjacent to S. Furthermore, for each j it holds that $w(V_j^{\top}) < S$, thus in total at most one copy of S is needed for S in the role of right cover square to cover the unoccupied space.

Claim 3: Each square is needed at most twelve times to cover unoccupied space. A square is used at most six times as left cover square and at most one time as right cover square. Furthermore, to cover the right trench an extra copy of the largest square is needed. Last of all, a square never has to cover more than one right piece, and this requires at most four copies of the square. Thus $6 + 1 + 1 + 4 = 12$ copies of the squares suffices to cover all the unoccupied space in the bottom-left packing restricted to the $[0, W] \times [0, h_{\mathrm{BL}} - 3h_{\max}]$ substrip. □

4.7 Boundedness for SSPP

This section exhibits the main result of this chapter. It is shown that the approximation ratio of the bottom-left algorithm for any ordering of squares is bounded by a constant. Therefore, even when the worst ordering of the squares is used, the approximation ratio remains bounded. This strongly contrasts the general case consisting of rectangles as shown in Theorem 4.1.

Although it might be interesting to get an approximation ratio as small as possible, this section only cares about boundedness. Theorem 4.33 expresses the bound in terms of abstract numbers, illustrating that improvement can be found by lowering any of the constants involved. Most certainly improvement is possible by either constructing a different covering or by enhanching the analysis of this chapter. Next, Corollary 4.34 substitutes the numbers found in the global cover theorem (Theorem 4.32) to obtain a 16-approximation.

Theorem 4.33. Let $\mathcal{I}$ be a `Square Strip Packing` instance, then

$$\frac{h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I})}{h_{\mathrm{OPT}}(\mathcal{I})} \leq f + g + 1.$$

Here f is the number of copies of the instance that is required to cover the unoccupied space in the $[0, W] \times [0, h_{\mathrm{BL}} - g h_{\max}]$ substrip for some constant g.

Proof. The total area of the strip $[0, W] \times [0, h_{\mathrm{BL}}]$ equals the total area of the squares A_{squares} plus the unoccupied area A_{unocc}. The unoccupied area is bounded by f copies of the squares plus the area outside of the $[0, W] \times [0, h_{\mathrm{BL}} - g h_{\max}]$ substrip. In other words, it holds that

$$A_{\mathrm{unocc}} \leq f A_{\mathrm{squares}} + g h_{\max} W.$$

Furthermore, the total area of the squares is bounded by the total area of an optimum packing, therefore it holds that

$$\begin{aligned} h_{\mathrm{BL}} W = A_{\mathrm{squares}} + A_{\mathrm{unocc}} &\leq (f+1) A_{\mathrm{squares}} + g h_{\max} W \\ &\leq (f+1) h_{\mathrm{OPT}} W + g h_{\max} W. \end{aligned}$$

It holds that $h_{\max} \leq h_{\mathrm{OPT}}$. Thus this implies that

$$h_{\mathrm{BL}} \leq (f+1) h_{\mathrm{OPT}} + g h_{\max} \leq (f + g + 1) h_{\mathrm{OPT}}.$$

□

The analysis of the previous proof actually gives an asymptotic approximation ratio, namely it shows that $h_{\mathrm{BL}} \leq (f+1) h_{\mathrm{OPT}} + g h_{\max}$. However, this is irrelevant for boundedness. Next, Corollary 4.34 uses the global cover theorem (Theorem 4.32) and Theorem 4.33 to show that the approximation ratio of an SSPP-instance using any order is bounded by 16.

Corollary 4.34. Let $\mathcal{I}$ be a `Square Strip Packing` instance, then

$$\frac{h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I})}{h_{\mathrm{OPT}}(\mathcal{I})} \leq 16.$$

Proof. The global cover theorem (Theorem 4.32) gives a covering of the unoccupied space in the $[0, W] \times [0, h_{\mathrm{BL}} - 3 h_{\max}]$ substrip of the bottom-left packing of an SSPP-instance using at most twelve copies of the squares from the instance. Hence, Theorem 4.33 implies that

$$\frac{h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I})}{h_{\mathrm{OPT}}(\mathcal{I})} \leq f + g + 1 = 12 + 3 + 1 = 16.$$

□

The `Online Square Strip Packing` problem (OSSPP) is a version where the squares arrive in a given order. Corollary 4.34 basically also shows that the bottom-left algorithm for OSSPP-instances has approximation ratio at most 16.

4.8 Lower bound

The 16-approximation ratio for the bottom-left algorithm using the worst-case ordering can almost certainly be improved by lowering the number of copies of squares that are required, making the cut-off substrip smaller, or by giving a tighter analysis. This section looks at how much the approximation ratio can be improved, by constructing a lower bound for the bottom-left algorithm using the worst ordering. The construction of the instance that results in the lower bound is based on the so-called checkerboard example as introduced by Baker et al. [BCR80]. The following example gives a demonstration of the checkerboard.

Example 4.35. Let $\varepsilon > 0$ be small, and let $m < n$ be natural numbers with m even. Consider the instance consisting of the squares of sizes $2 - i\varepsilon$ for $i = 1, \ldots, n$ together with $mn + (m-2)\frac{m}{2}$ squares of size 1. Let the width of the strip be $W = 2n - \frac{n(n+1)}{2}\varepsilon$. The bottom-left packing of the instance ordered by decreasing size is depicted in Figure 21. This bottom-left packing is called the (m, n)-*checkerboard*.

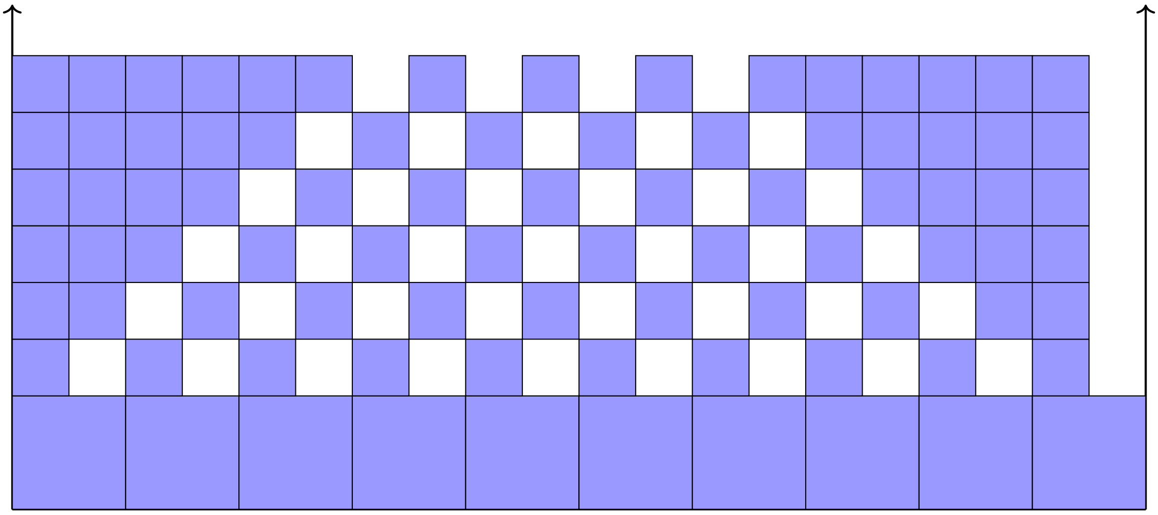


Figure 21: The checkerboard example from Baker et al. [BCR80].

All the large squares fit on the bottom of the strip, because it holds that

$$\sum_{i=1}^{n}(2 - i\varepsilon) = 2n - \frac{n(n+1)}{2}\varepsilon = W.$$

Since the large squares are ordered by decreasing size, it follows that the unit squares on top of the large squares are placed from right to left. Since two unit squares have larger width than a large square and squares are placed according to the bottom-left rule, it follows that only one unit square is placed on each large square. This type of assignment repeats in the second row since the holes between the squares in the first row all have width less than 1. In general it holds that the i-th row of unit squares alternates holes and squares except for the initial and final squares of the row. Now, each row contains n unit squares, namely one above each large square, further, the i-th odd row contains $i - 1$ extra squares in the initial and final part and the i-th even row contains $i-2$ such squares. Thus in m rows, the amount of units squares is

$$\begin{aligned}\sum_{i=1,\ i \text{ odd}}^{m} (n+i-1) + \sum_{i=1,\ i \text{ even}}^{m} (n+i-2) &= mn + \sum_{i=1}^{m}(i-1) - \sum_{i=1,\ i \text{ even}}^{m} 1 \\ &= mn + \frac{(m-1)m}{2} - \frac{m}{2} \\ &= mn + (m-2)\frac{m}{2}.\end{aligned}$$

In other words, all the unit squares fit into m rows. Hence the bottom-left packing as depicted in Figure 21 is correct. Thus the height of this bottom-left packing equals $m+2-\varepsilon$.

Suppose that $(m-2)\frac{m}{2} < W$, then there is a packing where the mn unit squares are placed into $\frac{m}{2}$ rows followed by one row consisting of the $(m-2)\frac{m}{2}$ unit squares, followed by the large squares. Thus an optimum packing has height at most $\frac{m}{2}+1+2$. Hence for large values of n and m, it follows that the approximation ratio of the bottom-left algorithm for this instance ordered by decreasing size converges to 2.

The construction for the lower bound of the worst-case bottom-left algorithm consists of a checkerboard followed by a repeating pattern of squares that form one row, followed by one big square on top of that row. This construction results in a $6-2\sqrt{2}$ lower bound.

Theorem 4.36 (Lower bound)**.** There exists a SSPP-instance $\mathcal{I}$ such that

$$\frac{h_{\mathrm{BL}}^{\mathrm{worst}}(\mathcal{I})}{h_{\mathrm{OPT}}(\mathcal{I})} \geq 6 - 2\sqrt{2} \approx 3.17.$$

Proof. Let $\varepsilon > 0$ be small, let $L > 0$ be such that $m = 6L\frac{1+\sqrt{2}}{2+\sqrt{2}}$ is an even integer and let n be such that $(m-2)\frac{m}{2} < W$ where $W = 2n - \frac{n(n+1)}{2}\varepsilon$. Define $p = \left\lfloor \frac{W}{(2+\sqrt{2})L} \right\rfloor$. Consider a (m,n)-checkerboard, together with at most $\lfloor (1+\sqrt{2})L \rfloor p$ squares of size 1, p squares of size L and exactly one square of size $(1+\sqrt{2})L$.

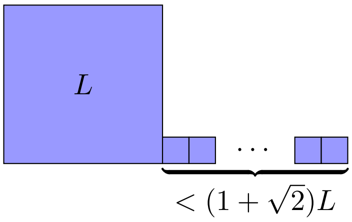


Figure 22: Pattern that is placed on top of the checkerboard. It consists of one square of size L on the left, and less than $(1+\sqrt{2})L$ unit squares and holes of unit width on the right.

The idea of the worst-case instance is to first place the checkerboard. Then, on top of this, there are placed p patterns as depicted in Figure 22. Finally, on top of these patterns there is placed one square of size $(1+\sqrt{2})L$. Then the total height of the bottom-left packing is

$$\begin{aligned}h_{\mathrm{BL}} &= (2-\varepsilon) + m + L + (1+\sqrt{2})L \\ &= (2-\varepsilon) + 6L\frac{1+\sqrt{2}}{2+\sqrt{2}} + (2+\sqrt{2})L.\end{aligned}$$

Placing the patterns turns out to be the most complicated. The patterns are placed from right to left just as the unit squares in the checkerboard were placed from right to left. In the middle of the checkerboard the unit squares alternate with holes as before, but for the sides of the checkerboard more unit squares are needed. The main idea of the pattern is to place less than $(1+\sqrt{2})L$ unit squares and holes followed by one square of size L. Placing p patterns means that the entire top of the checkerboard is covered with patterns. As the big holes between the patterns have width strictly less than $(1+\sqrt{2})L$, it follows that the last square of size $(1+\sqrt{2})L$ is placed on top of these patterns. Observe, that also the width between the left strip boundary and the left-most pattern should be less than $(1+\sqrt{2})L$, this can be ensured by varying the amount of unit squares in the patterns. This results in the packing as illustrated in Figure 23.

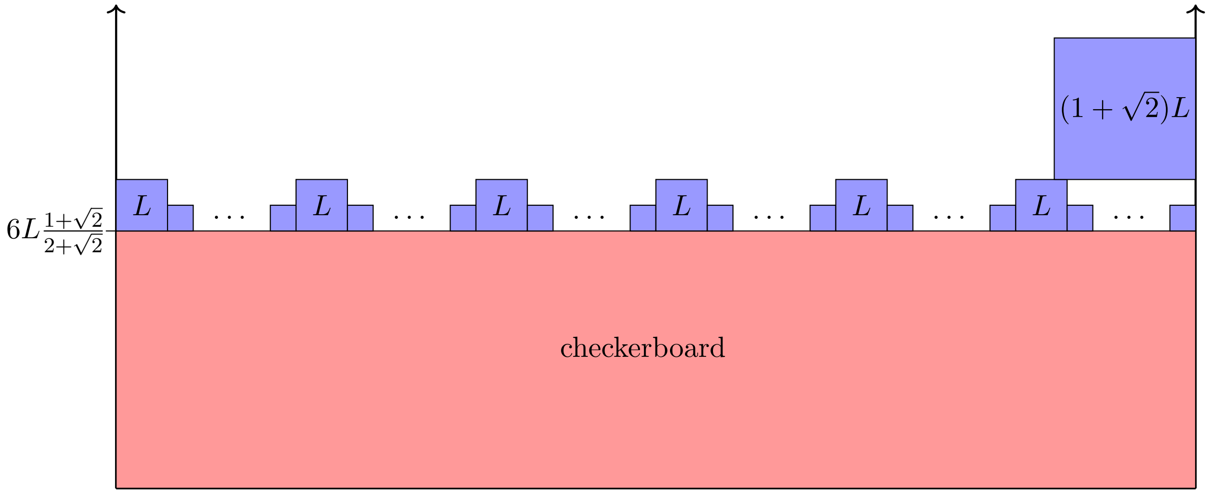


Figure 23: The worst-case bottom-left packing of the instance attaining the lower bound.

Now, for large L and n, the height of an optimum packing is $(1+\sqrt{2})L+\mathcal{O}(1)$. Namely, the amount of unoccupied space in the $[0,W]\times[0,(1+\sqrt{2})L]$ substrip after placing all the squares of size L is

$$\begin{aligned}(1+\sqrt{2})LW - pL &\geq (1+\sqrt{2})LW - \frac{W}{(2+\sqrt{2})L}L\\ &= LW\left(1+\sqrt{2}-\frac{1}{(2+\sqrt{2})L}\right)\\ &= LW\frac{(1+\sqrt{2})(2+\sqrt{2})L-1}{(2+\sqrt{2})L}\\ &= LW\frac{(4+3\sqrt{2})L-1}{(2+\sqrt{2})L}\\ &= 3LW\frac{1+\sqrt{2}}{2+\sqrt{2}}+\frac{L-1}{(2+\sqrt{2})L}.\end{aligned}$$

As there are at most $nm+(m-2)\frac{m}{2}+(1+\sqrt{2})Lp$ unit squares, it follows that the unit squares $nm = 6Ln\frac{1+\sqrt{2}}{2+\sqrt{2}}$ also fit in the $[0,W]\times[0,(1+\sqrt{2})L]$ substrip for sufficiently small ε, since it holds that $W=2n-\mathcal{O}(\varepsilon)$. The other unit squares $(m-2)\frac{m}{2}+(1+\sqrt{2})Lp$ fit into a constant number of rows on top of the $[0,W]\times[0,(1+\sqrt{2})L]$ substrip. The same holds for the squares $2-i\varepsilon$ for $i=1,\dots,n$. Finally, the square of size $(1+\sqrt{2})L$ can be placed in the $[0,W]\times[0,(1+\sqrt{2})L]$ substrip at the expense of an extra constant number of rows consisting of unit squares on top of the substrip. There exists an unbounded sequence of triples (L,n,ε)

such that $m = 6L\frac{1+\sqrt{2}}{2+\sqrt{2}}$ is an even integer and $(m-2)\frac{m}{2} < W$. Now, it holds that as $L \to \infty$, $n \to \infty$ and $\varepsilon \to 0$, the approximation ratio of the bottom-left packing of the constructed instance tends to

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}} \geq \lim_{L\to\infty} \frac{6L\frac{1+\sqrt{2}}{2+\sqrt{2}} + (2+\sqrt{2})L + \mathcal{O}(1)}{(1+\sqrt{2})L + \mathcal{O}(1)} = \frac{6}{2+\sqrt{2}} + \frac{2+\sqrt{2}}{1+\sqrt{2}} = 6 - 2\sqrt{2}.$$

□

5 Better-than-2 bottom-left algorithms for SSPP

In Chapter 4 it was demonstrated that the approximation ratio of the bottom-left algorithm remains bounded irrespective of the order of the squares. Currently, the best-known approximation ratio achieved by the bottom-left algorithm for `Square Strip Packing` is 2, which is attained by ordering the squares by decreasing size [BCR80].

Contrary to ordering the squares by decreasing size, Section 5.1 considers ordering the squares by increasing size. The bottom-left packing of squares by increasing size has an interesting structure, namely, the squares are packed in rows. Section 5.1 proves that the approximation ratio of the bottom-left algorithm for squares ordered by increasing size is at most 3. Although this is worse than the 2-approximation of Baker et al. [BCR80], there are conditions for which the approximation ratio is better than 2. These conditions are explored in Section 5.2. First of all, when the ratio between the width of the strip and the size of the largest square tends to infinity, the approximation ratio converges to 2. Moreover, Section 5.3 shows that in the limit this is tight. Secondly, the analysis of the performance of the bottom-left algorithm for squares ordered by increasing size gives an asymptotic approximation ratio. Consequently, when the largest square is considerably smaller than the height of the optimum packing, the approximation ratio attained by the bottom-left algorithm is also better than 2.

Furthermore, in Section 5.4 a new order is constructed such that for each instance the height of the bottom-left packing is smaller or equal to the height of the bottom-left packing when the squares are ordered by increasing size. In other words, using this newly constructed ordering of the squares is always better than using the increasing ordering. The analysis of this section will heavily rely on the results of the increasing size ordering. It will also be shown that this new ordering is a tight 2-approximation and is better-than-2 in many special cases.

5.1 Ordering squares by increasing size

Analoguously to Chapter 4, the first step is to analyze the structure of a bottom-left packing where the squares are placed in increasing order of size. It turns out that in such a packing, the squares are packed in rows. Subsequently, this structure is used to bound the amount of unoccupied space in the bottom-left packing, eventually resulting in a 3-approximation for the bottom-left algorithm. After establishing this 3-approximation, Section 5.2 shows that as the ratio of the width of the strip and the size of the largest square goes to infinity, the approximation ratio goes to 2. Moreover, Section 5.3 constructs a sequence of instances such that the ratio $\frac{W}{h_{\max}}$ converges to infinity and the approximation ratio converges to 2.

Theorem 5.1 (Structure theorem)**.** Let $\mathcal{I}_{\text{incr}} = (\{S_1, \dots, S_n\}, W)$ be a SSPP-instance ordered by increasing size. Apply the bottom-left algorithm to this instance. Then, there are sets of squares $\mathcal{R}_i = \{S^i_1, \dots, S^i_{n_i}\}$ for $1 \leq i \leq \ell$, called *rows*, such that

(a) The rows partition the squares of the instance $\mathcal{I}$.

(b) It holds that $S^1_1 <_{\text{BL}} \cdots <_{\text{BL}} S^1_{n_1} <_{\text{BL}} S^2_1 <_{\text{BL}} \cdots <_{\text{BL}} S^\ell_{n_\ell}$.

(c) For every $1 \leq i \leq \ell$, the left face of S^i_1 is adjacent to the left strip boundary and the right face of S^i_j is adjacent to the left face of S^i_{j+1} for $1 \leq j < n_i$.

Remark 5.2. The squares are ordered by increasing size, hence for two different squares S_a and S_b, it holds that $S_a <_{\text{BL}} S_b$ implies $S_a \leq S_b$, and $S_a < S_b$ implies $S_a <_{\text{BL}} S_b$. Thus Theorem 5.1 part (b) implies that

$$S^1_1 \leq \cdots \leq S^1_{n_1} \leq S^2_1 \leq \cdots \leq S^\ell_{n_\ell}.$$

The row $\mathcal{R}_1$ is the lowest row and $\mathcal{R}_i$ is the i-th row counted from below. There are ℓ rows. Furthermore, part (c) implies that S^i_j is the j-th square in the i-th row. There are n_i squares in row i.

Proof of Theorem 5.1 by induction on n. The base case is trivial, for $\mathcal{I} = (\{S_1\}, W)$ just define one row $\mathcal{R}_1 = \{S_1\}$. Obviously this satisfies (a), (b) and (c).

Next assume that the theorem is true for instances of size n. For the induction step, consider an instance $\mathcal{I}_{\text{incr}} = (\{S_1, \dots, S_{n+1}\}, W)$ consisting of $n+1$ squares that are ordered by increasing size. Let $\mathcal{R}'_1, \dots, \mathcal{R}'_\ell$ be the rows of the subinstance $\mathcal{I}'_{\text{incr}} = (\{S_1, \dots, S_n\}, W)$ that exist by the induction hypothesis. Notice that $\text{tf}(S^\ell_1) \le \text{tf}(S^\ell_{n_\ell})$, because $S^\ell_1 <_{\text{BL}} S^\ell_{n_\ell}$ and $S^\ell_1 \le S^\ell_{n_\ell}$ by (b).

Claim: The substrip $B = [\text{rf}(S^\ell_{n_\ell}), W] \times [\text{tf}(S^{\ell-1}_{n_{\ell-1}}), \text{tf}(S^{\ell-1}_{n_{\ell-1}}) + (W - \text{rf}(S^\ell_{n_\ell}))]$ is unoccupied after placing the squares in the subinstance $\mathcal{I}'_{\text{incr}}$. Define $\text{tf}(S^0_{n_0}) = 0$. The set B is depicted in Figure 24. To prove the claim, it is shown that no square intersects B (up to measure 0). Indeed there are no squares in $\mathcal{R}_\ell$ to the right of $S^\ell_{n_\ell}$ by (c), hence no square in row ℓ intersects B. Furthermore, for each square $S \in \mathcal{R}_1 \cup \dots \cup \mathcal{R}_{\ell-1}$ it holds that $\text{tf}(S) \le \text{tf}(S^{\ell-1}_{n_{\ell-1}})$, because it holds that $S \le S^{\ell-1}_{n_{\ell-1}}$ and S is placed before $S^{\ell-1}_{n_{\ell-1}}$, so if $\text{tf}(S) > \text{tf}(S^{\ell-1}_{n_{\ell-1}})$, then S could have been placed lower. Thus also nothing in row $1, \dots, \ell-1$ intersects B. This proves the claim.

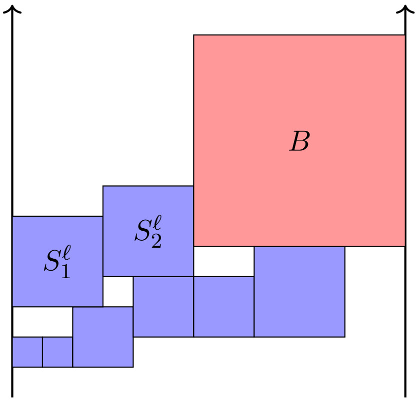


Figure 24: Part of a bottom-left packing of squares ordered by increasing size. The substrip B is depicted in red.

The square S_{n+1} is placed after S_n and $S_n \le S_{n+1}$, hence it must hold that $S_n <_{\text{BL}} S_{n+1}$. Thus S_{n+1} is placed above the line $\text{bf}(S^\ell_{n_\ell})$ by the bottom-left algorithm. Now distinguish two cases: either S_{n+1} is placed next to $S^\ell_{n_\ell}$ or S_{n+1} is placed in a new row.

Case 1: Suppose that $S_{n+1} \le W - \text{rf}(S^\ell_{n_\ell})$. Then S_{n+1} fits into the unoccupied space B. It holds that $\text{bf}(B) = \text{tf}(S^{\ell-1}_{n_{\ell-1}}) \le \text{tf}(S^\ell_i)$ for each $1 \le i \le n_\ell$, because $S^{\ell-1}_{n_{\ell-1}}$ is placed before S^ℓ_i and $S^{\ell-1}_{n_{\ell-1}} \le S^\ell_i$, so otherwise $S^{\ell-1}_{n_{\ell-1}}$ could have been placed lower. Thus it follows that the bottom-left algorithm places S_{n+1} next to $S^\ell_{n_\ell}$. Define $\mathcal{R}_i = \mathcal{R}'_i$ for $i < \ell$ and $\mathcal{R}_\ell = \mathcal{R}'_\ell \cup \{S_{n+1}\}$. Clearly this satisfies (a), (b) and (c).

Case 2: Suppose that $S_{n+1} > W - \text{rf}(S^\ell_{n_\ell})$. Then S_{n+1} does not fit in B, hence it does not fit next to $S^\ell_{n_\ell}$. By (b) the top faces of the squares in the ℓ-th row are increasing in y-coordinate. Thus as S_n must be bottom-left of S_{n+1}, it follows that S_{n+1} is placed against the left strip boundary above the ℓ-th row. Thus define $\mathcal{R}_i = \mathcal{R}'_i$ for $i \le \ell$ and $\mathcal{R}_{\ell+1} = \{S_{n+1}\}$. This partition also satisfies (a), (b) and (c). $\square$

Remark 5.3 (ℓ-approximation). Suppose that the bottom-left packing of squares ordered by increasing size consists of ℓ rows. Then the bottom-left packing has approximation ratio at most ℓ. Indeed, each row increases the height of the packing by at most $h_{\max}$ and $h_{\max} \leq h_{\text{OPT}}$, thus it holds that

$$h_{\text{BL}}^{\text{incr}} \leq \ell h_{\max} \leq \ell h_{\text{OPT}}.$$

Each bottom face of a rectangle in a bottom-left packing is adjacent to either the bottom strip boundary or another square. Otherwise the rectangle could have been placed closer to the bottom. This leads to the definition of pillar squares, which will be crucial in the analysis of the approximation ratio.

Definition 5.4. A rectangle P that is placed adjacent to the bottom face of a rectangle S is called a *pillar rectangle* or *pillar square* of S.

In case of increasing size squares, the structure theorem implies that each square in row i has a pillar square in row $i-1$. Hence a sequence of pillar squares from the top square all the way to the bottom of the strip can be defined.

Definition 5.5. Consider the bottom-left packing of a SSPP-instance $\mathcal{I}_{\text{incr}}$. Define $S_{p_\ell}^{\ell} = S_{n_\ell}^{\ell}$ and inductively define $S_{p_i}^{i}$ to be the left-most pillar square of $S_{p_{i+1}}^{i+1}$ for $1 \leq i < \ell$. The sequence $S_{p_1}^{1}, \ldots, S_{p_\ell}^{\ell}$ is called the *(left-most) pillar sequence* of $\mathcal{I}_{\text{incr}}$. Similarly, the *right-most pillar sequence* can be defined.

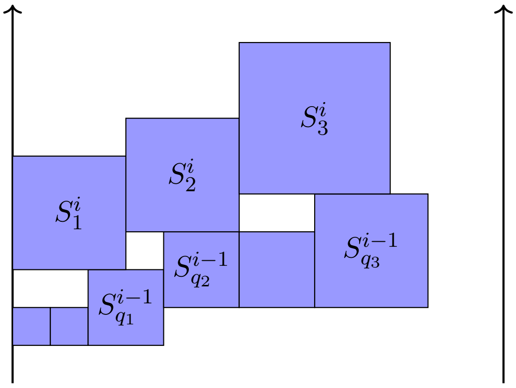


Figure 25: Structure of a bottom-left packing where squares are ordered by increasing size. Here $S_{q_j}^{i-1}$ is the left-most pillar square of S_j^i.

Figure 25 illustrates the concept of rows when the squares are placed by increasing size. Moreover, the left-most pillar squares are indicated. Suppose that a square has two pillar squares S_a and S_b, then these squares have the same size, because without loss of generality assume $S_a <_{\text{BL}} S_b$, then it follows that $S_a \leq S_b$, and it holds that $\text{tf}(S_a) = \text{tf}(S_b)$, thus this implies that $S_a = S_b$. The following lemma is a crucial observation about the pillar sequence.

Lemma 5.6. Let $S_{p_1}^{1}, \ldots, S_{p_\ell}^{\ell}$ be the (left-most or right-most) pillar sequence of SSPP-instance $\mathcal{I}_{\text{incr}}$. Then it holds that $\sum_{i=1}^{\ell} S_{p_i}^{i} = h_{\text{BL}}$.

Proof. Observe that $\text{tf}(S_{p_i}) = \text{bf}(S_{p_{i+1}}^{i+1})$. It follows that

$$\sum_{i=1}^{\ell} S_{p_i}^{i} = \sum_{i=1}^{\ell} \text{tf}(S_{p_i}^{i}) - \text{bf}(S_{p_i}^{i}) = \text{tf}(S_{p_\ell}^{\ell}) - \text{bf}(S_{p_1}^{1}) = h_{\text{BL}} - 0 = h_{\text{BL}}.$$

□

To show that the bottom-left algorithm for increasing size squares has approximation ratio 3, it first has to be shown that the amount of unoccupied space is bounded. The structure theorem (Theorem 5.1) implies that each vertical line in the substrip $[0, W - h_{\max}] \times [0, h_{\text{BL}}]$ of the packing intersects either $\ell - 1$ or ℓ squares. This fact together with the pillar sequence is the basis for the following vertical cover theorem.

Theorem 5.7 (Vertical cover theorem)**.** Let $\mathcal{I}_{\text{incr}}$ be a SSPP-instance, and let ℓ_v be a vertical line in the substrip $[0, W - h_{\max}] \times [0, h_{\text{BL}}]$ of the bottom-left packing. Then ℓ_v intersects at most $2h_{\max}$ unoccupied space and at most $h_{\max}$ unoccupied space if ℓ_v is to the left of $\text{rf}(S^{\ell}_{n_\ell})$.

Proof. For $1 \leq i < \ell$ denote by u_i the amount of unoccupied space between row i and row $i+1$ intersecting ℓ_v, and denote by u_ℓ the amount between row ℓ and h_{BL} intersecting ℓ_v. The total amount of unoccupied space intersecting the vertical line ℓ_v is $\sum_{i=1}^{\ell} u_i$. Furthermore, denote by $S^i_{\ell_v}$ a square in row i intersecting ℓ_v. Distinguish two cases: ℓ_v left or right of $\text{rf}(S^{\ell}_{n_\ell})$.

Case 1: The vertical line ℓ_v is to the right of the right face of $S^{\ell}_{n_\ell}$. Then ℓ_v intersects $\ell - 1$ squares and it holds that

$$u_1 + \sum_{i=2}^{\ell-1} (u_i + S^i_{\ell_v}) + u_\ell \leq h_{\text{BL}} = \sum_{i=1}^{\ell} S^i_{p_i} \leq \sum_{i=2}^{\ell-1} S^i_{\ell_v} + 2h_{\max}$$

The first term is less than or equal to the length of ℓ_v, which is h_{BL}. The height h_{BL} equals the sum of the heights of the pillar sequence squares by Lemma 5.6. Furthermore, the last inequality uses that $S^i_{p_i} \leq S^{i+1}_{\ell_v}$ and $S^{\ell-1}_{p_{\ell-1}} \leq S^{\ell}_{p_\ell} = h_{\max}$. It follows that

$$\sum_{i=1}^{\ell} u_i \leq 2h_{\max}.$$

Case 2: The vertical line ℓ_v is to the left of the right face of $S^{\ell}_{n_\ell}$. Then ℓ_v intersects ℓ squares and it holds that

$$u_1 + \sum_{i=2}^{\ell} (u_i + S^i_{\ell_v}) \leq h_{\text{BL}} = \sum_{i=1}^{\ell} S^i_{p_i} \leq \sum_{i=2}^{\ell} S^i_{\ell_v} + h_{\max}$$

Now, it follows that

$$\sum_{i=1}^{\ell} u_i \leq h_{\max}.$$

□

The next theorem uses the vertical cover theorem (Theorem 5.7) to obtain that the bottom-left algorithm for squares ordered by increasing size has approximation ratio 3. The analysis of the Theorem 5.8 is similar to the proof of Theorem 4.33 that showed boundedness of the approximation ratio of bottom-left packing for squares using the worst ordering.

Theorem 5.8 (3-approximation)**.** For a SSPP-instance $\mathcal{I}_{\text{incr}}$ it holds that $h^{\text{incr}}_{\text{BL}}(\mathcal{I}) \leq 3h_{\text{OPT}}(\mathcal{I})$.

Proof. The total area of the strip $[0, W] \times [0, h_{\text{BL}}]$ equals the total area of the squares A_{squares} plus the unoccupied area A_{unocc}. The vertical cover theorem (Theorem 5.7) shows that the maximal amount of unoccupied space in the $[0, W - h_{\max}] \times [0, h_{\text{BL}}]$ substrip is attained when $S^{\ell}_{n_\ell}$ is placed against the left strip boundary. Integrating over unoccupied space intersecting vertical lines in this substrip gives at most $h_{\max}^2 + (W - 2h_{\max})2h_{\max}$ unoccupied space. Furthermore, the substrip $[W - h_{\max}, W] \times [0, h_{\mathrm{BL}}]$ gives at most $h_{\mathrm{BL}}h_{\max}$ unoccupied space. All in all, this implies that

$$\begin{aligned} h_{\mathrm{BL}}W = A_{\mathrm{squares}} + A_{\mathrm{unocc}} &\leq h_{\mathrm{OPT}}W + h_{\max}^2 + (W - 2h_{\max})2h_{\max} + h_{\mathrm{BL}}h_{\max} \\ &= h_{\mathrm{OPT}}W + (2W - 3h_{\max})h_{\max} + h_{\mathrm{BL}}h_{\max}. \end{aligned}$$

Thus it follows that

$$h_{\mathrm{BL}} \leq \frac{W}{W - h_{\max}} h_{\mathrm{OPT}} + \frac{2W - 3h_{\max}}{W - h_{\max}} h_{\max} \leq 3h_{\mathrm{OPT}},$$

as $h_{\max} \leq h_{\mathrm{OPT}}$. □

Remark 5.9. Observe that the analysis of Theorem 5.8 requires that $h_{\max} < W$. This requirement can always be satisfied by removing squares of size W, as those only make the approximation ratio smaller, since their contribution to h_{BL} (for any ordering of the squares) equals their contribution to h_{OPT}. Actually, it is possible to remove all squares of size larger than $W - h_{\min}$, because also their contribution to h_{BL} equals their contribution to h_{OPT}.

5.2 Asymptotic approximation ratio depending on $k = \frac{W}{h_{\max}}$

Instead of constructing a cover using vertical lines, this section constructs a cover using horizontal lines. For an instance $\mathcal{I}$, define the ratio between the width of the strip and the height of the largest square as $k = \frac{W}{h_{\max}}$. It is shown that as k goes to infinity, the approximation ratio of the bottom-left algorithm on increasing squares converges to 2. Moreover, it is shown that if the height of the largest square is relatively small to the height of an optimum packing, then the approximation ratio becomes better than 2.

The vertical cover theorem argued that for each vertical line a constant number of squares was intersected plus or minus one. For the horizontal cover theorem a similar argument lies at its heart, namely, each horizontal line intersects at most two rows. The fact that a horizontal line intersects at most two rows implies that the amount of unoccupied space is limited. The following lemma states that each row climbs at most $S_1^i + S_{n_i}^i$, that is, $\mathrm{tf}(S_{n_i}^i) \leq \mathrm{bf}(S_1^i) + S_1^i + S_{n_i}^i$, This is a direct consequence of the structure theorem.

Lemma 5.10. Consider the bottom-left packing of SSPP-instance $\mathcal{I}_{\mathrm{incr}}$. Then for each $1 \leq i \leq \ell$ and $1 \leq j \leq n_i$ it holds that $\mathrm{bf}(S_j^i) \leq \mathrm{tf}(S_1^i)$.

Proof. Assume $i \geq 2$, as otherwise $\mathrm{bf}(S_j^i) = \mathrm{bf}(S_1^i) < \mathrm{tf}(S_1^i)$ for each $1 \leq i \leq n_1$. Let $S_{q_j}^{i-1}$ be a pillar square of S_j^i, then (b) from the structure theorem (Theorem 5.1) implies that $S_{q_j}^{i-1} \leq S_1^i$. Thus indeed it holds that

$$\mathrm{bf}(S_j^i) = \mathrm{tf}(S_{q_j}^{i-1}) = \mathrm{bf}(S_{q_j}^{i-1}) + S_{q_j}^{i-1} \leq \mathrm{bf}(S_1^i) + S_1^i = \mathrm{tf}(S_1^i).$$

□

The horizontal substrips $[0, W] \times [\mathrm{bf}(S_{p_i}^i), \mathrm{tf}(S_{p_i}^i)]$ for $1 \leq i \leq \ell$ partition the area of the bottom-left packing by definition of the pillar sequence $S_{p_1}^1, \ldots, S_{p_\ell}^\ell$. As Lemma 5.10 shows that rows only go up slowly, it follows that the amount of unoccupied space in such a substrip can be bounded. This is the main result of the next theorem.

Theorem 5.11 (Horizontal cover theorem)**.** Let $\mathcal{R}_i = \{S^i_1, \dots, S^i_{n_i}\}$ for $1 \leq i \leq \ell$ be the partition in rows of a SSPP-instance $\mathcal{I}_{\text{incr}}$ and let $S^1_{p_1}, \dots, S^\ell_{p_\ell}$ be the left-most pillar sequence. Define A^i_{unocc} as the amount of unoccupied space in the $[0, W] \times [\text{bf}(S^i_{p_i}), \text{tf}(S^i_{p_i})]$ substrip of the bottom-left packing. Then it holds that

$$\begin{aligned} A^i_{\text{unocc}} &\leq (S^{i+1}_{p_{i+1}} + S^{i+1}_1) S^i_{p_i}, \quad \text{for } 1 \leq i < \ell, \\ A^\ell_{\text{unocc}} &\leq (W - h_{\max}) h_{\max}. \end{aligned}$$

Proof. For $1 \leq i < \ell$ and $1 \leq j \leq n_{i+1}$, denote by $S^i_{q_j}$ and $S^i_{r_j}$ the left-most respectively right-most pillar square of S^{i+1}_j. Let S^{i+1}_0 be the formal square that is placed at $\{0\} \times [\text{tf}(S^i_1), \text{tf}(S^{i+1}_1)]$ and denote by S^0_j the bottom strip boundary. Observe that $S^i_{r_0} = S^i_1$. In the adjacency graph of the bottom-left packing of the instance $\mathcal{I}_{\text{incr}}$, let $P = S^i_{t_1}, \dots, S^i_{t_s}$ be the unique path in row i consisting of right-type arrows from $S^i_{t_1} = S^i_{r_{j-1}}$ to $S^i_{t_s} = S^i_{q_j}$. This unique path exists by (c) of the structure theorem (Theorem 5.1). Also consider the path $Q = S^i_{r_{j-1}}, S^{i+1}_{j-1}, S^{i+1}_j, S^i_{q_j}$ consisting of an up-type arrow, a right-type arrow and a down-type arrow. The undirected union of the paths P and Q forms a cycle surrounding unoccupied space, as illustrated in Figure 26. Define U^i_j to be the unoccupied space enclosed by $P \cup Q$. Observe that possibly $U^i_j = \varnothing$, for example if $S^i_{r_{j-1}} = S^i_{q_j}$. Denote by A^i_j the area of U^i_j.

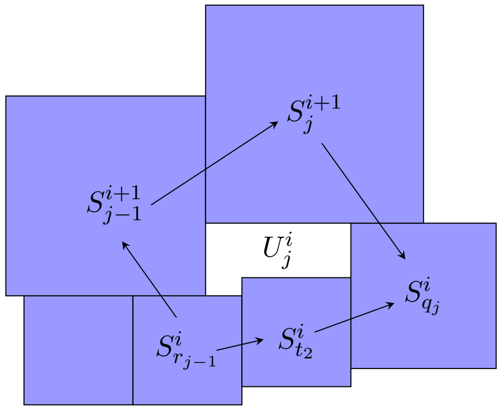


Figure 26: The path $P = S^i_{r_{j-1}}, S^i_{t_2}, S^i_{q_j}$ and the path $Q = S^i_{r_{j-1}}, S^{i+1}_{j-1}, S^{i+1}_j, S^i_{q_j}$. The unoccupied space enclosed by $P \cup Q$ is denoted by U^i_j.

Claim 1: It holds that $A^i_j \leq S^{i+1}_j(\text{bf}(S^{i+1}_j) - \text{tf}(S^i_{r_{j-1}}))$ for $1 \leq i < \ell$. To prove this claim, notice that almost all the arrows in $P \cup Q$ around U^i_j are of right-type, except $(S^i_{r_{j-1}}, S^{i+1}_{j-1})$ and $(S^{i+1}_j, S^i_{q_j})$. Furthermore, it holds that $\text{rf}(S^i_{r_{j-1}}) \geq \text{rf}(S^{i+1}_{j-1})$, because $S^i_{r_{j-1}}$ is the right-most pillar square of S^{i+1}_{j-1}. Similarly, $\text{lf}(S^i_{q_j}) \leq \text{rf}(S^{i+1}_j)$, because $S^i_{q_j}$ is the left-most pillar square of S^{i+1}_j. These three facts together imply that the width of U^i_j is bounded by

$$\text{lf}(S^i_{q_j}) - \text{rf}(S^{i+1}_{j-1}) \leq \text{rf}(S^{i+1}_j) - \text{lf}(S^{i+1}_j) = S^{i+1}_j.$$

The path P is below U^i_j and part (b) of the structure theorem (Theorem 5.1) implies that $\text{tf}(S^i_{t_k}) \leq \text{tf}(S^i_{t_{k'}})$ for $k \leq k'$. This implies that U^i_j is above $\text{tf}(S^i_{t_1}) = \text{tf}(S^i_{r_{j-1}})$. The path Q is above U^i_j, with only $(S^{i+1}_{j-1}, S^{i+1}_j)$ as right-type arrow. Therefore, $\text{bf}(S^{i+1}_{j-1}) \leq \text{bf}(S^{i+1}_j)$ implies that U^i_j is below $\text{bf}(S^{i+1}_j)$. Thus the height of U^i_j is bounded by $\text{bf}(S^{i+1}_j) - \text{tf}(S^i_{r_{j-1}})$. This proves the claim.

Claim 2: For $1 \le i < \ell$, the unoccupied space in the $B_i = [0, W - S_1^{i+1}] \times [\mathrm{bf}(S_{p_i}^i), \mathrm{tf}(S_{p_i}^i)]$ substrip is contained in

$$\bigcup_{j=1}^{p_{i+1}} U_j^i \cup \bigcup_{j=p_i+1}^{n_i} U_j^{i-1}.$$

To prove this claim, notice that the top faces of the squares $S_1^{i+1}, \dots, S_{p_{i+1}}^{i+1}$ are strictly above the top face of $S_{p_i}^i$. Thus there cannot be unoccupied space above those squares intersecting with B_i. Furthermore, the bottom faces of the squares $S_1^i, \dots, S_{p_i-1}^i$ are below the bottom face of $S_{p_i}^i$, hence there cannot be unoccupied space below these squares intersecting with B_i. The only unoccupied space directly below S_j^{i+1} in the substrip B_i is U_j^i by definition. Thus the unoccupied space left of $S_{p_i}^i$ in the substrip B_i equals $\bigcup_{j=1}^{p_{i+1}} U_j^i$.

A similar argument holds for the unoccupied space right of $S_{p_i}^i$ in the substrip B_i. The top faces of the squares $S_{p_i+1}^i, \dots, S_{n_i}^i$ are above the top face of $S_{p_i}^i$ and the bottom faces of the squares $S_{p_{i-1}}^{i-1}, \dots, S_{n_{i-1}}^{i-1}$ are strictly below the bottom face of $S_{p_i}^i$. The width of the substrip B_i equals $W - S_1^{i+1}$, therefore all the unoccupied space right of $S_{p_i}^i$ in the substrip B_i is directly under the squares $S_{p_i+1}^i, \dots, S_{n_i}^i$. In other words, this unoccupied space is contained in $\bigcup_{j=p_i+1}^{n_i} U_j^{i-1}$, but not necessarily equal to it. This proves the claim.

Claim 3: For $1 \le i < \ell$ it holds that $A_{\mathrm{unocc}}^i \le (S_{p_{i+1}}^{i+1} + S_1^{i+1}) S_{p_i}^i$. To prove this claim, observe that the unoccupied space in the substrip $[0, W] \times [\mathrm{bf}(S_{p_i}^i), \mathrm{tf}(S_{p_i}^i)]$ equals the unoccupied space in B_i together with the unoccupied space in $[W - S_1^{i+1}, W] \times [\mathrm{bf}(S_{p_i}^i), \mathrm{tf}(S_{p_i}^i)]$. Claim 1 and 2 together imply that the amount of unoccupied space in B_i is bounded by

$$\begin{aligned}
\sum_{j=1}^{p_{i+1}} A_j^i + \sum_{j=p_i+1}^{n_i} A_j^{i-1} &\le \sum_{j=1}^{p_{i+1}} S_j^{i+1}(\mathrm{bf}(S_j^{i+1}) - \mathrm{bf}(S_{j-1}^{i+1})) + \sum_{j=p_i+1}^{n_i} S_j^i(\mathrm{bf}(S_j^i) - \mathrm{bf}(S_{j-1}^i)) \\
&\le S_{p_{i+1}}^{i+1} \sum_{j=1}^{p_{i+1}} (\mathrm{bf}(S_j^{i+1}) - \mathrm{bf}(S_{j-1}^{i+1})) + S_{p_{i+1}}^{i+1} \sum_{j=p_i+1}^{n_i} (\mathrm{bf}(S_j^i) - \mathrm{bf}(S_{j-1}^i)) \\
&= S_{p_{i+1}}^{i+1}(\mathrm{bf}(S_{p_{i+1}}^{i+1}) - \mathrm{tf}(S_1^i) + \mathrm{bf}(S_{n_i}^i) - \mathrm{bf}(S_{p_i}^i)) \\
&\le S_{p_{i+1}}^{i+1} S_{p_i}^i.
\end{aligned}$$

The first inequality follows from Claim 1 together with $\mathrm{tf}(S_{r_{j-1}}^i) = \mathrm{bf}(S_{j-1}^{i+1})$. The second inequality uses that $S_j^{i+1} \le S_{p_{i+1}}^{i+1}$ for $1 \le j \le p_{i+1}$ and $S_j^i \le S_{p_{i+1}}^{i+1}$ for $p_i + 1 \le j \le n_i$. Next, the sums are telescoping. The final inequality uses that $\mathrm{bf}(S_{n_i}^i) \le \mathrm{tf}(S_1^i)$ as shown in Lemma 5.10, together with the fact that $\mathrm{bf}(S_{p_{i+1}}^{i+1}) - \mathrm{bf}(S_{p_i}^i) = \mathrm{tf}(S_{p_i}^i) - \mathrm{bf}(S_{p_i}^i) = S_{p_i}^i$. Furthermore, the unoccupied space in $[W - S_1^{i+1}, W] \times [\mathrm{bf}(S_{p_i}^i), \mathrm{tf}(S_{p_i}^i)]$ is bounded by the total area, which has width S_1^{i+1} and height $S_{p_i}^i$. This proves the claim.

Claim 4: It holds that $A_{\mathrm{unocc}}^\ell \le (W - h_{\max}) h_{\max}$. The total area in $[0, W] \times [\mathrm{bf}(S_{p_\ell}^\ell), \mathrm{tf}(S_{p_\ell}^\ell)]$ equals $W h_{\max}$. At least $S_{p_\ell}^\ell$ is contained in this substrip, thus the total unoccupied area is bounded by $W h_{\max} - h_{\max}^2$. Now, the theorem is proven by Claim 3 and 4. □

In the special case where $k = \frac{W}{h_{\max}}$ is larger than 3, the horizontal cover theorem implies a better-than-3 approximation ratio for the bottom-left algorithm for squares ordered by increasing size. The analysis of the better-than-3 approximation is similar to the analysis of the 3-approximation that resulted from the vertical cover theorem.

Theorem 5.12 (Better-than-3 approximation). Let $\mathcal{I}$ be a `Square Strip Packing` instance ordered by increasing size. Assume that $k = \frac{W}{h_{\max}} > 2$. Then

$$h_{\text{BL}}^{\text{incr}} \leq \frac{k}{k-2} h_{\text{OPT}} + \frac{k-3}{k-2} h_{\max}.$$

Proof. The total area of the strip $[0, W] \times [0, h_{\text{BL}}]$ equals the total area of the squares A_{squares} plus the unoccupied area A_{unocc}. Now, by definition of the pillar sequence $S_{p_1}^1, \ldots, S_{p_\ell}^\ell$, it holds that $A_{\text{unocc}} = \sum_{i=1}^{\ell} A_{\text{unocc}}^i$. Thus the horizontal cover theorem (Theorem 5.11) implies that

$$\begin{aligned} h_{\text{BL}} W &\leq h_{\text{OPT}} W + \sum_{i=1}^{\ell-1} (S_{p_{i+1}}^{i+1} + S_1^{i+1}) S_{p_i}^i + (W - h_{\max}) h_{\max} \\ &\leq h_{\text{OPT}} W + 2 h_{\max} \sum_{i=1}^{\ell-1} S_{p_i}^i + (W - h_{\max}) h_{\max} \\ &= h_{\text{OPT}} W + 2 h_{\max} (h_{\text{BL}} - h_{\max}) + (W - h_{\max}) h_{\max}. \end{aligned}$$

Hence it follows that

$$h_{\text{BL}} \leq \frac{W}{W - 2h_{\max}} h_{\text{OPT}} + \frac{W - 3h_{\max}}{W - 2h_{\max}} h_{\max}.$$

□

Using that $h_{\max} \leq h_{\text{OPT}}$, the approximation ratio of Theorem 5.12 tends to 2 as k goes to infinity, because in the limit $h_{\text{BL}} \leq h_{\text{OPT}} + h_{\max} \leq 2h_{\text{OPT}}$. Also observe that if $h_{\max} \leq \frac{1}{p} h_{\text{OPT}}$ for some $p \geq 1$, then the approximation ratio also improves to $\frac{(p+1)k-3}{p(k-2)}$. Hence for $k > \frac{4p-3}{p-1}$, the approximation ratio is better-than-2. For example, if $p = 2$, then the approximation ratio is better-than-2 for all $k > 5$.

These two findings strongly differ from the 2-approximation ratio for the bottom-left algorithm when squares are ordered by decreasing size. The lower bound for this 2-approximation that Baker et al. [BCR80] construct, called the checkerboard example (Example 4.35), shows that regardsless of when $k \to \infty$ or $p \to \infty$, the approximation ratio remains 2. Thus in cases where $k = \frac{W}{h_{\max}}$ and $p = \frac{h_{\text{OPT}}}{h_{\max}}$ are large, the approximation ratio of the bottom-left algorithm for squares ordered by increasing size beats the approximation ratio for squares ordered by decreasing size in the worst-case.

5.3 Tightness in the limit, $k \to \infty$

Section 5.1 showed that the approximation ratio for the bottom-left algorithm applied to squares ordered by increasing size is at most 3. Moreover, Section 5.2 showed that for $k = \frac{W}{h_{\max}} > 2$, it even holds that

$$h_{\text{BL}}^{\text{incr}} \leq \frac{k}{k-2} h_{\text{OPT}} + \frac{k-3}{k-2} h_{\max}.$$

This section discusses the tightness of this result. In particular, an instance is constructed where the bottom-left algorithm for increasing size squares has approximation ratio $\frac{5}{2}$. This is the worst ratio found, leaving a gap of $\frac{1}{2}$ between the lower bound $\frac{5}{2}$ and the upper bound 3. Furhtermore, tightness in the limit is shown, that is, a sequence of instances is constructed that shows that if $k \to \infty$ the approximation ratio approaches 2.

Example 5.13. Let $r \geq 3$ be an integer, $h > 0$ a large number and $\varepsilon > 0$ a small number such that $\frac{2rh}{\varepsilon}$ is an integer. Consider the instance consisting of $\frac{2rh}{\varepsilon} + (r-1)(r-2)$ squares of size ε, $(r-1)(r-2)$ squares of size $\frac{r}{r-1}h - \varepsilon$, $r-1$ squares of size $\frac{r}{r-1}h$ and $r-1$ squares of size $rh+1$. Let the width of the strip equal $W = r(rh+1) - 1 - \varepsilon$. The bottom-left algorithm applied to the instance ordered by increasing size is depicted in Figure 27.

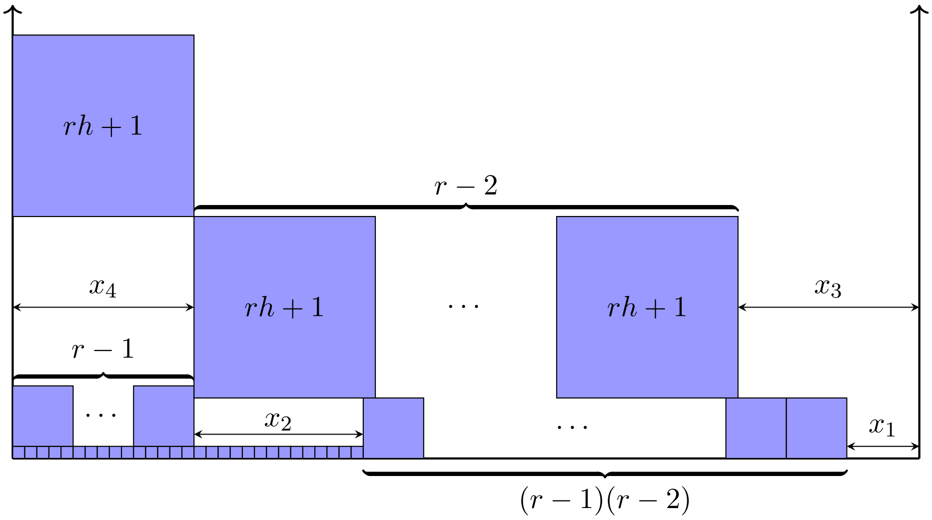


Figure 27: A bottom-left packing of instance consisting of squares ordered by increasing size.

To see that this packing is correct the distances x_1, x_2, x_3 and x_4 are computed. The first row consists of all squares of size ε and all squares of size $\frac{r}{r-1}h - \varepsilon$, this fits in one row since

$$\left(\frac{2rh}{\varepsilon} + (r-1)(r-2)\right)\varepsilon + (r-1)(r-2)\left(\frac{r}{r-1}h - \varepsilon\right) = 2rh + (r-2)rh = r^2h < W.$$

The remaining space right of the first row is

$$x_1 = W - r^2h = r(rh+1) - 1 - \varepsilon - r^2h = r - 1 - \varepsilon.$$

Since it holds that $\frac{r}{r-1}h > x_1$ for sufficiently large h, a second row is created. The second row consists of all squares of size $\frac{r}{r-1}h$ and of $r-2$ squares of size $rh+1$. These squares fit next to each other in one row since

$$(r-1)\frac{r}{r-1}h + (r-2)(rh+1) = (r-1)(rh+1) - 1 < W.$$

The squares of size $\frac{r}{r-1}h$ fit on top of the squares of size ε, but the squares of size $rh+1$ are placed above the squares of size $\frac{r}{r-1}h - \varepsilon$, since for sufficiently small ε it holds that

$$\begin{aligned} x_2 &= \left(\frac{2rh}{\varepsilon} + (r-1)(r-2)\right)\varepsilon - (r-1)\frac{r}{r-1}h \\ &= 2rh + (r-1)(r-2)\varepsilon - rh = rh + (r-1)(r-2)\varepsilon \\ &< rh + 1. \end{aligned}$$

The last square of size $rh+1$ forms a third row, because it does not fit to the right of the second row, as

$$\begin{aligned} x_3 &= W - (r-1)\frac{r}{r-1}h - (r-2)(rh+1) \\ &= r(rh+1) - 1 - \varepsilon - rh - (r-2)(rh+1) \\ &= 2(rh+1) - 1 - \varepsilon - rh = rh + 1 - \varepsilon < rh + 1. \end{aligned}$$

And the square of size $rh+1$ does not fit on top of the squares of size $\frac{r}{r-1}h$, because

$$x_4 = (r-1)\frac{r}{r-1}h = rh < rh + 1.$$

Thus the height of the bottom-left packing using increasing size equals $h_{\text{BL}} = 2(rh+1) + \frac{r}{r-1}h - \varepsilon$.

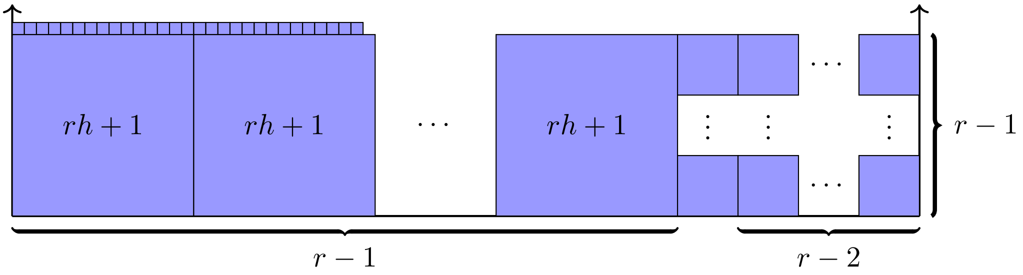


Figure 28: A feasible packing of the squares in the instance with height $rh + 1 + \varepsilon$.

Next, a packing of height $rh + 1 + \varepsilon$ is depicted in Figure 28. This packing has the $r - 1$ squares of size $rh + 1$ packed in a row on the left. Then there are $r - 1$ columns, the first column consists of $r - 1$ squares of size $\frac{r}{r-1}h$, the other $r - 2$ columns consists of $r - 1$ squares of size $\frac{r}{r-1}h - \varepsilon$. Finally, all the squares of size ε are placed on top of the squares of size $rh+1$. First of all, notice that all the squares from the instance are packed. Secondly, notice that this is a feasible packing as

$$\begin{aligned} (r-1)(rh+1) + \frac{r}{r-1}h + (r-2)\left(\frac{r}{r-1}h - \varepsilon\right) &= (r-1)(rh+1) + (r-1)\frac{r}{r-1}h - (r-2)\varepsilon \\ &= r(rh+1) - 1 - (r-2)\varepsilon < W. \end{aligned}$$

Thus the squares in the packing fit next to each other. All in all, the instance has approximation ratio

$$\frac{h_{\text{BL}}}{h_{\text{OPT}}} \geq \lim_{h\to\infty}\lim_{\varepsilon\to 0} \frac{2(rh+1) + \frac{r}{r-1}h - \varepsilon}{rh + 1 + \varepsilon} = 2 + \frac{1}{r-1}.$$

Notice that $k = \frac{r(rh+1)-1-\varepsilon}{rh+1} = r - \frac{1+\varepsilon}{rh+1}$ and $k = r$ if $h \to \infty$ and $\varepsilon \to 0$. This shows that as $k = \frac{W}{h_{\max}}$ tends to infinity, the approximation ratio goes to 2. However, this does not prove that the approximation ratio from Theorem 5.12 for the bottom-left algorithm applied to squares ordered by increasing size is tight for every value of k.

Table 1 shows for different values of $k = \frac{W}{h_{\max}}$ the proven upper and lower bounds of the approximation ratio of the bottom-left algorithm applied to instances consisting of squares that are ordered by increasing size.

k	upper bound	lower bound
3	3	$2\frac{1}{2}$
4	$2\frac{1}{2}$	$2\frac{1}{3}$
5	$2\frac{1}{3}$	$2\frac{1}{4}$
6	$2\frac{1}{4}$	$2\frac{1}{5}$
7	$2\frac{1}{5}$	$2\frac{1}{6}$

Table 1: Upper bound of approximation ratio for bottom-left algorithm applied to squares ordered by increasing size according to Theorem 5.12. And lower bound according to Example 5.13.

5.4 The last-row-full ordering

The vertical cover theorem (Theorem 5.7) states that vertical line right of the right face of the last placed square $S^\ell_{n_\ell}$ intersect at most $2h_{\max}$ unoccupied space and only at most $h_{\max}$ unoccupied space when the vertical line is left of the right face of $S^\ell_{n_\ell}$. In the analysis of the 3-approximation of the bottom-left algorithm for squares ordered by increasing size (Theorem 5.8) it was assumed that $S^\ell_{n_\ell}$ is adjacent to the left strip boundary, in other words, the last row consists of one square, namely of the square $S^\ell_{n_\ell}$. It is clear that if $S^\ell_{n_\ell}$ is more to the right, then the vertical cover theorem gives a better approximation ratio.

The same is true for the top row in the horizontal cover theorem (Theorem 5.11), it was shown that $A^\ell_{\text{unocc}} \leq (W - h_{\max})h_{\max}$. In other words, the amount of unoccupied space in the top row was bounded by a large number, because the top row can be almost entirely unoccupied, as illustrated in Example 5.13.

This section discusses a new order similar to increasing size where all rows are "full" and a few small squares are placed in the end. It will be shown in Theorem 5.16 that for each instance, the height of the bottom-left packing of this new ordering is better than the height of the bottom-left packing when the squares are ordered by increasing size. Furthermore, Theorem 5.18 will show that the bottom-left algorithm using this new ordering has approximation ratio equal to 2 and is better-than-2 in many special cases.

Definition 5.14. Consider an instance $\mathcal{I}$ of `Square Strip Packing`. The *last-row-full* (LRF) order is defined as the order that is returned by Algorithm 2.

Algorithm 2 Last-row-full ordering

```
Input: SSPP-instance $\mathcal{I}_{\text{incr}} = (\mathcal{S}_{\text{incr}}, W)$ where $\mathcal{S}_{\text{incr}} = \{S_1, \ldots, S_n\}$
Output: the last-row-full ordering of $\mathcal{I}$
$w \leftarrow W$, $\ell \leftarrow 1$
for $i = n, \ldots, 1$ do
    if $w - S_i \geq 0$ then
        $\mathcal{Q}_\ell \leftarrow \{S_i\} \cup \mathcal{Q}_\ell$
        $w \leftarrow w - S_i$
    else
        $\ell \leftarrow \ell + 1$
        $\mathcal{Q}_\ell \leftarrow \{S_i\}$
        $w \leftarrow W - S_i$
    end if
end for
return the order $\mathcal{Q}_{\ell-1}, \mathcal{Q}_{\ell-2}, \ldots, \mathcal{Q}_1, \mathcal{Q}_\ell$ where each $\mathcal{Q}_j$ is ordered by increasing size
```

The next lemma states that the structure in rows is still preserved for the large squares, and that only the small squares are placed differently. Moreover, it states that for every row, the last square of the previous row cannot be placed in front of the row.

Lemma 5.15. For an instance $\mathcal{I}$ let $\mathcal{Q}_{\ell-1}, \ldots, \mathcal{Q}_1, \mathcal{Q}_\ell$ be the order returned by Algorithm 2. Let $\mathcal{R}_i = \mathcal{Q}_{\ell-i} = \{S_1^i, \ldots, S_{m_i}^i\}$ for $1 \leq i < \ell$ and $\mathcal{R}_{\text{small}} = \mathcal{Q}_\ell = \{S_{t_1}, \ldots, S_{t_s}\}$. Then it holds that $S_{t_1} \leq \cdots \leq S_{t_s} \leq S_1^1 \leq \cdots \leq S_{m_1}^1 \leq \cdots \leq S_{m_{\ell-1}}^{\ell-1}$ and it holds that

$$\sum_{j=t_1}^{t_s} S_j \leq W \quad \text{and} \quad \sum_{j=1}^{m_i} S_j^i \leq W < S_{m_{i-1}}^{i-1} + \sum_{j=1}^{m_i} S_j^i$$

for all $i = 1, \ldots, \ell-1$, where $S_{m_{-1}}^{-1} = S_{t_s}$.

Proof. Each $\mathcal{Q}_j$ is ordered by increasing size, hence it holds that $S_1^i \leq \cdots \leq S_{m_i}^i$ for each $1 \leq i < \ell$ and $S_{t_1} \leq \cdots \leq S_{t_s}$. Furthermore, the for-loop runs over the squares in decreasing order and fills the $\mathcal{Q}_j$ in the order $\mathcal{Q}_1, \ldots, \mathcal{Q}_\ell$. The algorithm returns the $\mathcal{Q}_j$ in the order $\mathcal{Q}_{\ell-1}, \ldots, \mathcal{Q}_1, \mathcal{Q}_\ell$, hence it follows that

$$S_{t_1} \leq \cdots \leq S_{t_s} \leq S_1^1 \leq \cdots \leq S_{m_1}^1 \leq \cdots \leq S_{m_{\ell-1}}^{\ell-1}.$$

Last of all, the if-else-statement implies that the index ℓ is increased once the row $\mathcal{Q}_i$ is full. Thus it holds that $\sum_{j=1}^{m_i} S_j^i \leq W$ for each $1 \leq i < \ell$. Moreover, it also holds that $\sum_{j=t_1}^{t_s} S_j \leq W$. Furthermore, this also implies that adding $S_{m_{i-1}}^{i-1}$ to the squares in $\mathcal{Q}_i$ would exceed the width of the strip. That is, it holds that $S_{m_{i-1}}^{i-1} + \sum_{j=1}^{m_i} S_j^i > W$ for all $1 \leq i < \ell$. □

As the squares $S_1^1 \leq \cdots \leq S_{m_1}^1 \leq S_1^2 \leq \cdots \leq S_{m_{\ell-1}}^{\ell-1}$ are ordered by increasing size, it follows that these squares satisfy the structure theorem (Theorem 5.1). Intuitively, the other squares $S_{t_1} \leq \cdots \leq S_{t_s}$ have a small effect on the height of the packing, because their size is smaller than the squares that were packed before. The small squares that are placed in the end will either fill up unoccupied space of the packing of the larger squares, or will be placed on top of the larger squares.

It turns out that the LRF-ordering is always better than the increasing size ordering. The next theorem proves this by comparing two pillar sequences, one in the increasing size packing and one in the last-row-full bottom-left packing.

Theorem 5.16. For each SSPP-instance $\mathcal{I}$, it holds that $h_{\text{BL}}^{\text{LRF}}(\mathcal{I}) \leq h_{\text{BL}}^{\text{incr}}(\mathcal{I})$.

Proof. Let $\mathcal{J}$ be the subinstance of $\mathcal{I}$ consisting of the squares $S_1^1, \ldots, S_{m_1}^1, \ldots, S_{m_{\ell-1}}^{\ell-1}$. For the bottom-left packing of $\mathcal{I}_{\text{incr}}$ ordered by increasing size, let $S_{p_1}^1, \ldots, S_{p_\ell}^\ell$ be the right-most pillar sequence. Similar, for the bottom-left packing of $\mathcal{J}_{\text{incr}}$ ordered by increasing size, let $S_{r_1}^1, \ldots, S_{r_{\ell-1}}^{\ell-1}$ be the right-most pillar sequence.

Claim: For $i = 1, \ldots, \ell-1$ it holds that $S_{r_i}^i \leq S_{p_{i+1}}^{i+1}$ and $S_{r_i}^i$ is the last square in the i-th row in the bottom-left packing of $\mathcal{J}_{\text{incr}}$. To prove this claim, use induction on i, starting with $i = \ell-1$ and working downwards. For $i = \ell-1$ the statement holds because $S_{r_{\ell-1}}^{\ell-1} = h_{\max} = S_{p_\ell}^\ell$ and $S_{r_{\ell-1}}^{\ell-1}$ is the last square of the top row by definition of $\mathcal{J}_{\text{incr}}$.

Next assume the claim holds for all $j > i$. Then $S_{r_{i+1}}^{i+1}$ is the last square in the $(i+1)$-th row in the bottom-left packing of $\mathcal{J}_{\text{incr}}$. Now suppose that $S_{r_i}^i$ is not the last square in the i-th row in the bottom-left packing of $\mathcal{J}_{\text{incr}}$, then there is a different square $S_{m_i}^i$ after $S_{r_i}^i$ in this row. However, as $S_{r_i}^i$ is the right-most pillar square of $S_{r_{i+1}}^{i+1}$ and $S_{r_{i+1}}^{i+1}$ is the last square in row $i+1$, it follows that $S_{m_i}^i + \sum_{j=1}^{m_{i+1}} S_j^{i+1} \leq W$. This contradicts Lemma 5.15, hence $S_{r_i}^i$ must be the last square in the i-th row.

It remains to show that $S^i_{r_i} \leq S^{i+1}_{p_{i+1}}$. Let $S^i_{q_{r_{i+1}}}$ be the right-most pillar square of $S^{i+1}_{r_{i+1}}$ in the bottom-left packing of $\mathcal{I}_{\text{incr}}$. By the induction hypothesis it holds that $S^{i+1}_{r_{i+1}} \leq S^{i+2}_{p_{i+2}}$, hence in the bottom-left packing of $\mathcal{I}_{\text{incr}}$ it also holds that their pillar squares have this relation, that is, $S^i_{q_{r_{i+1}}} \leq S^{i+1}_{p_{i+1}}$. Denote the sum of the widths of the squares after $S^i_{q_{r_{i+1}}}$ in the i-th row by w_1 and the sum of the widths of the squares before $S^{i+1}_{r_{i+1}}$ in the $(i+1)$-th row of the bottom-left packing of $\mathcal{I}_{\text{incr}}$ by w_2, this is illustrated in Figure 29. It holds that $w_1 + w_2 + S^{i+1}_{r_{i+1}} \leq W$, therefore, all the squares after $S^i_{q_{r_{i+1}}}$ in the i-th row of the bottom-left packing of $\mathcal{I}_{\text{incr}}$ will be placed in the $(i+1)$-th row of the bottom-left packing of $\mathcal{J}_{\text{incr}}$ because $S^{i+1}_{r_{i+1}}$ is the last square in this row. Hence it follows that $S^i_{r_i}$ is before (or equal to) $S^i_{q_{r_{i+1}}}$ in the bottom-left packing of $\mathcal{I}_{\text{incr}}$. Therefore $S^i_{r_i} \leq S^i_{q_{r_{i+1}}} \leq S^{i+1}_{p_{i+1}}$. Thus the claim is true by induction.

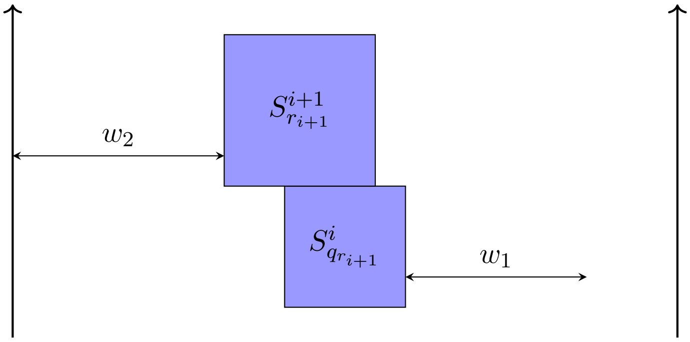


Figure 29: Illustration of $S^{i+1}_{r_{i+1}}$ and its right-most pillar square $S^i_{q_{r_{i+1}}}$ in the bottom-left packing of $\mathcal{I}_{\text{incr}}$. The sum of the widths of the squares after $S^i_{q_{r_{i+1}}}$ in row i equals w_1 and the sum of the widths of the squares before $S^{i+1}_{r_{i+1}}$ in row $i+1$ equals w_2. Observe that $w_1 + w_2 + S^{i+1}_{r_{i+1}} \leq W$.

Let $S_{t_1} \leq \cdots \leq S_{t_s}$ be the squares in $\mathcal{I} \setminus \mathcal{J}$ ordered by increasing size. These squares are placed last in the bottom-left packing using the LRF-ordering. Now suppose it holds that $h^{\text{LRF}}_{\text{BL}}(\mathcal{I}) > h^{\text{incr}}_{\text{BL}}(\mathcal{J}) + S_{t_s}$, then at least two of the squares in $\mathcal{I} \setminus \mathcal{J}$ are stacked on top of each other with one strictly above $h^{\text{incr}}_{\text{BL}}(\mathcal{J})$. However, this is impossible, as then all the space above the last square $S^{\ell-1}_{m_{\ell-1}}$ in the packing of $\mathcal{J}_{\text{incr}}$ must be occupied. This can only happen if there is a row consisting of the squares $S_{t_1}, \ldots, S_{t_s}$ from the left strip boundary to the space above $S^{\ell-1}_{m_{\ell-1}}$. But as $\sum_{j=1}^{s} S_{t_j} \leq W$ holds by Lemma 5.15, after this there are no squares left. This contradiction implies that it must hold that $h^{\text{LRF}}_{\text{BL}}(\mathcal{I}) \leq h^{\text{incr}}_{\text{BL}}(\mathcal{J}) + S_{t_s}$.

Using the claim, it holds that $S_{t_s} \leq S^1_{p_1}$, because $S^2_{r_2} \leq S^2_{p_2}$, implies the same relation for the right-most pillar squares in the bottom-left packing of $\mathcal{I}_{\text{incr}}$, that is, $S^1_{q_{r_2}} \leq S^1_{p_1}$. As all the squares after $S^1_{q_{r_2}}$ are packed in the first row of the packing of $\mathcal{J}_{\text{incr}}$, it follows that S_{t_s} is left of (or equal to) $S^1_{q_{r_2}}$ in the bottom-left packing of $\mathcal{I}_{\text{incr}}$. Therefore it holds that $S_{t_s} \leq S^1_{q_{r_2}} \leq S^1_{p_1}$.

In conclusion, it holds that

$$h^{\text{LRF}}_{\text{BL}}(\mathcal{I}) \leq h^{\text{incr}}_{\text{BL}}(\mathcal{J}) + S_{t_s} \leq \sum_{i=1}^{\ell-1} S^i_{r_i} + S^1_{p_1} \leq \sum_{i=1}^{\ell} S^i_{p_i} = h^{\text{incr}}_{\text{BL}}(\mathcal{I}).$$

□

Notice that Theorem 5.16 is true regardless of the order of the small squares $S_{t_1}, \ldots, S_{t_s}$, as the fact that these squares were ordered by increasing size was never used in the proof. Therefore, it is best to order the squares $S_{t_1}, \ldots, S_{t_s}$ by decreasing size. Because then the smallest squares do not unnecessarily use large unoccupied space that is better occupied by the larger small squares.

As mentioned in the begin of this section, intuitively, the LRF-ordering should give a better approximation ratio then the increasing size ordering, because more space in the top of the packing is occupied. Indeed, the following horizontal cover theorem (Theorem 5.17) gives a bound on the total amount of unoccupied space. Afterwards, it is shown that the LRF-ordering establishes that the bottom-left algorithm has approximation ratio equal to 2.

Theorem 5.17 (Horizontal cover theorem)**.** Let $\mathcal{I}_{\text{LRF}}$ be an SSPP-instance. Then the total amount of unoccupied space in the bottom-left packing of $\mathcal{I}_{\text{LRF}}$ is bounded by

$$A_{\text{unocc}} \leq (W - h_{\max})h_{\max} + h_{\max}(h_{\text{BL}} - h_{\max}).$$

Proof. Let $\mathcal{R}_i = \{S_1^i, \ldots, S_{m_i}^i\}$ for $1 \leq i \leq \ell$ and $\mathcal{R}_{\text{small}} = \{S_{t_1}, \ldots, S_{t_s}\}$ be the rows of the bottom-left packing of $\mathcal{I}_{\text{LRF}}$ as described in Lemma 5.15. Assume that $S_{m_0}^0$ is the largest square in $\mathcal{R}_{\text{small}}$. It is possible to adjust the bottom-left packing to a shelf packing such that the height of the packings are the same. That is, consider the packing where each square $S \in \mathcal{R}_i$ for $1 \leq i \leq \ell$ has the same x-coordinate, but it is lifted such that the y-coordinate equals the y-coordinate of the bottom face of $S_{m_i}^i$ in the bottom-left packing. This is possible because the last square $S_{m_i}^i$ in row i also belongs to the pillar sequence, as demonstrated in the proof of Theorem 5.16. Furthermore, ignore the small squares except for $S_{m_0}^0 \in \mathcal{R}_{\text{small}}$. The square $S_{m_0}^0$ is the largest square in $\mathcal{R}_{\text{small}}$ and is also the last square placed, therefore either $\text{tf}(S_{m_0}^0) = h_{\text{BL}}$ or everything in $\mathcal{R}_{\text{small}}$ is placed strictly below h_{BL}. Now place $S_{m_0}^0$ at the same position as in the bottom-left packing, hence it might overlap with another square, meaning that the constructed shelf packing is formally not an orthogonal packing. However, the amount of unoccupied area of the shelf packing is an upper bound for the amount of unoccupied area of the bottom-left packing. Figure 30 depicts the shelf packing of a bottom-left packing.

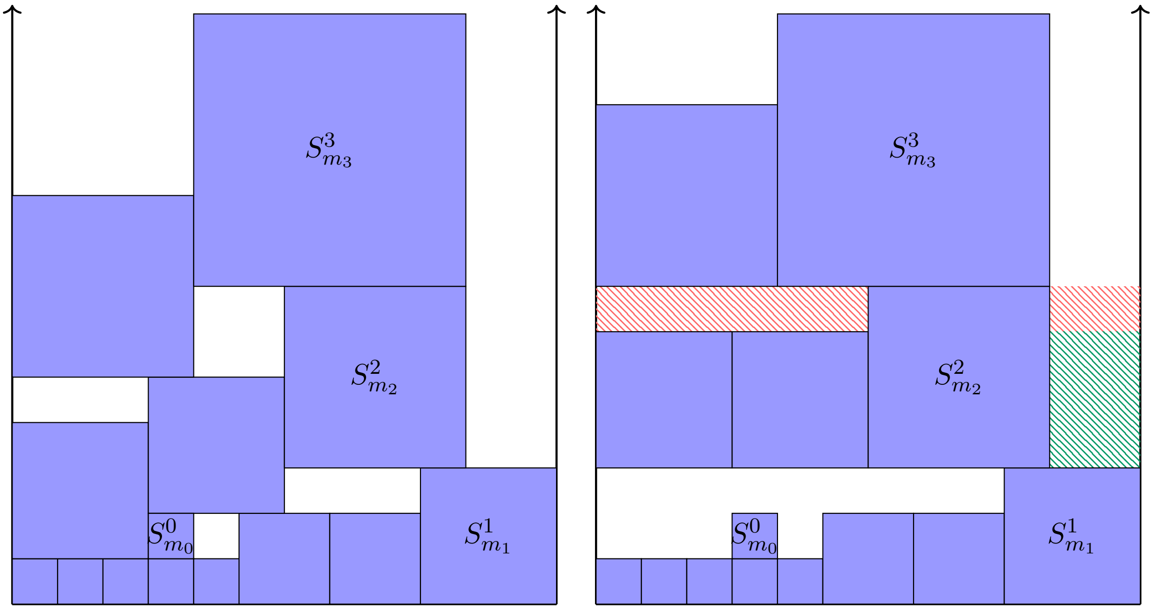


Figure 30: The bottom-left packing of the squares $1, 1, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 6, 1$ (on the left), together with its shelf packing (on the right). The unoccupied area A_{unocc}^2 consists of the red and the green area. The red area is bounded by $(W - S_{m_2}^2)(S_{m_2}^2 - S_{m_1}^1)$ and the green area is bounded by $(S_{m_1}^1)^2$.

Define A^i_{unocc} as the amount of unoccupied space in the $[0, W] \times [\text{bf}(S^i_{m_i}), \text{tf}(S^i_{m_i})]$ substrip of the shelf packing for $1 \leq i \leq \ell$. Furthermore let A^0_{unocc} be the amount of unoccupied space in the $[0, W] \times [\text{tf}(S^\ell_{m_\ell}), h_{\text{BL}}]$ substrip of the shelf packing.

Claim 1: For $1 \leq i \leq \ell$ it holds that $A^i_{\text{unocc}} \leq (W - S^i_{m_i})(S^i_{m_i} - S^{i-1}_{m_{i-1}}) + (S^{i-1}_{m_{i-1}})^2$. The amount of unoccupied space in the $[0, W] \times [\text{bf}(S^i_{m_i}), \text{bf}(S^i_{m_i}) + S^{i-1}_{m_{i-1}}]$ is bounded by $(S^{i-1}_{m_{i-1}})^2$, because the height equals $S^{i-1}_{m_{i-1}}$ and the width is bounded by $S^{i-1}_{m_{i-1}}$ since by Lemma 5.15 it holds that

$$W - \sum_{j=1}^{m_i} S^i_j \leq S^{i-1}_{m_{i-1}}.$$

Furthermore, the amount of unoccupied space in the $[0, W] \times [\text{bf}(S^i_{m_i}) + S^{i-1}_{m_{i-1}}, \text{tf}(S^i_{m_i})]$ substrip is bounded by $(W - S_{m_i})(S^i_{m_i} - S^{i-1}_{m_{i-1}})$, because the height equals $S^i_{m_i} - S^{i-1}_{m_{i-1}}$ and at least the square $S^i_{m_i}$ occupies the strip, thus the width of the unoccupied space is bounded by $W - S_{m_i}$. This is also demonstrated in Figure 30.

Claim 2: It holds that $A^0_{\text{unocc}} \leq (W - S^0_{m_0})(h_{\text{BL}} - \sum_{i=1}^{\ell} S^i_{m_i})$. At least the square $S^0_{m_0}$ is contained in the $[0, W] \times [\text{tf}(S^\ell_{m_\ell}), h_{\text{BL}}]$ substrip, hence the width of the unoccupied space is bounded by $W - S^0_{m_0}$. Furthermore, the height of the substrip is bounded by $h_{\text{BL}} - \sum_{i=1}^{\ell} S^i_{m_i}$, because $S^1_{m_1}, \ldots, S^\ell_{m_\ell}$ is a path in the up type adjacency graph from the strip bottom to $S^\ell_{m_\ell}$. This proves the claim.

Now, a bound on the total amount of unoccupied space in the bottom-left packing of $\mathcal{I}_{\text{LRF}}$ is given by

$$\begin{aligned}
\sum_{i=0}^{\ell} A^i_{\text{unocc}} &\leq (W - S^0_{m_0})(h_{\text{BL}} - \sum_{i=1}^{\ell} S^i_{m_i}) + \sum_{i=1}^{\ell} \Big((W - S^i_{m_i})(S^i_{m_i} - S^{i-1}_{m_{i-1}}) + (S^{i-1}_{m_{i-1}})^2\Big) \\
&= (W - S^0_{m_0})(h_{\text{BL}} - \sum_{i=1}^{\ell} S^i_{m_i}) + \sum_{i=1}^{\ell} W(S^i_{m_i} - S^{i-1}_{m_{i-1}}) \\
&\quad + \sum_{i=1}^{\ell} \Big((S^{i-1}_{m_{i-1}})^2 - (S^i_{m_i})^2\Big) + \sum_{i=1}^{\ell} S^i_{m_i} S^{i-1}_{m_{i-1}} \\
&= (W - S^0_{m_0})(h_{\text{BL}} - \sum_{i=1}^{\ell} S^i_{m_i}) + W(S^\ell_{m_\ell} - S^0_{m_0}) + (S^0_{m_0})^2 - (S^\ell_{m_\ell})^2 + \sum_{i=1}^{\ell} S^i_{m_i} S^{i-1}_{m_{i-1}} \\
&= (W - S^0_{m_0})(h_{\text{BL}} - \sum_{i=0}^{\ell} S^i_{m_i}) + (W - h_{\max})h_{\max} + \sum_{i=1}^{\ell} S^i_{m_i} S^{i-1}_{m_{i-1}}.
\end{aligned}$$

Define $y = \sum_{i=0}^{\ell} S^i_{m_i} - h_{\text{BL}}$ and observe that $0 \leq y \leq S^0_{m_0}$. It holds that

$$\begin{aligned}
(W - S^0_{m_0})(h_{\text{BL}} - \sum_{i=0}^{\ell} S^i_{m_i}) + \sum_{i=1}^{\ell} S^i_{m_i} S^{i-1}_{m_{i-1}} &\leq -(W - S^0_{m_0})y + S^1_{m_1} S^0_{m_0} + h_{\max} \sum_{i=2}^{\ell} S^{i-1}_{m_{i-1}} \\
&= -(W - S^0_{m_0})y + S^1_{m_1} y + S^1_{m_1}(S^0_{m_0} - y) + h_{\max} \sum_{i=2}^{\ell} S^{i-1}_{m_{i-1}} \\
&\leq (S^1_{m_1} - (W - S^0_{m_0}))y + h_{\max} \left(\sum_{i=1}^{\ell} S^{i-1}_{m_{i-1}} - y \right) \\
&= (S^1_{m_1} - (W - S^0_{m_0}))y + h_{\max} (h_{\text{BL}} - h_{\max}).
\end{aligned}$$

Now it holds that $(S^1_{m_1} - (W - S^0_{m_0}))y \leq 0$, because either $S^1_{m_1} \leq W - S^0_{m_0}$ in which case it is true, or $S^1_{m_1} > W - S^0_{m_0}$ in which case $S^1_{m_1}$ and $S^0_{m_0}$ do not fit on the same height in the packing. Therefore, $S^0_{m_0}$ does not fit on the same height as any $S^i_{m_i}$, thus $y = 0$, since $S^0_{m_0}$ must be packed above all the other $S^i_{m_i}$. All in all, the total unoccupied space in the bottom-left packing of $\mathcal{I}_{\text{LRF}}$ is bounded by

$$A_{\text{unocc}} \leq (W - h_{\max})h_{\max} + h_{\max}(h_{\text{BL}} - h_{\max}).$$

□

The horizontal cover theorem (Theorem 5.17) directly implies Theorem 5.18 stating that the bottom-left algorithm using the LRF-ordering is a 2-approximation. Moreover, this theorem actually result in an asymptotic approximation ratio depending on k. It can be seen that for many values of p and k, the absolute approximation ratio becomes better-than-2.

Theorem 5.18 (Better-than-2 approximation)**.** Let $\mathcal{I}_{\text{LRF}}$ be a SSPP-instance. Then

$$h^{\text{LRF}}_{\text{BL}} \leq \frac{k}{k-1}h_{\text{OPT}} + \frac{k-2}{k-1}h_{\max} \leq 2h_{\text{OPT}}.$$

Proof. The total area of the strip $[0, W] \times [0, h_{\text{BL}}]$ equals the total area of the squares plus the total unoccupied area. Hence, using the horizontal cover theorem (Theorem 5.17), it follows that

$$h_{\text{BL}}W \leq h_{\text{OPT}}W + (W - h_{\max})h_{\max} + h_{\max}(h_{\text{BL}} - h_{\max}).$$

This implies that

$$h_{\text{BL}}(W - h_{\max}) \leq h_{\text{OPT}}W + (W - 2h_{\max})h_{\max}.$$

Since $h_{\max} \leq h_{\text{OPT}}$, it holds that

$$h_{\text{BL}} \leq \frac{W}{W - h_{\max}}h_{\text{OPT}} + \frac{W - 2h_{\max}}{W - h_{\max}}h_{\max} \leq \frac{2W - 2h_{\max}}{W - h_{\max}}h_{\text{OPT}} = 2h_{\text{OPT}}.$$

□

In conclusion, the last-row-full ordering is better than the decreasing size ordering from Baker et al. [BCR80], because both orderings result in a 2-approximation, however, when $p = \frac{h_{\text{OPT}}}{h_{\max}} > 1$ and $k = \frac{W}{h_{\max}} > 2$, then the approximation ratio for the LRF-ordering becomes better-than-2 while the ratio for the decreasing size ordering remains 2. Namely, Theorem 5.18 gives an asymptotic approximation ratio, that results in the absolute approximation ratio

$$\frac{h_{\text{BL}}}{h_{\text{OPT}}} \leq \frac{k}{k-1} + \frac{1}{p}\frac{k-2}{k-1} = \frac{(p+1)k-2}{p(k-1)}.$$

However, there are instances where $p \approx 1$ and k goes to infinity such that the approximation ratio approaches 2. In other words, the approximation ratio of 2 for the bottom-left algorithm using the LRF-ordering is tight.

Theorem 5.19. For every $\varepsilon > 0$, there exists a SSPP-instance such that $h_{\mathrm{BL}}^{\mathrm{LRF}} \geq (2-\varepsilon)h_{\mathrm{OPT}}$.

Proof. Consider the instance consisting of a strip of width W together with $\frac{W}{\varepsilon^2} + \frac{W-1}{\varepsilon} + 1$ squares of size ε and one square of size 1. Assume that $\varepsilon > 0$ is chosen such that $\frac{1}{\varepsilon}$ is an integer. The bottom-left packing of the instance using the LRF ordering consists of $\frac{1}{\varepsilon}$ full rows of $\frac{W}{\varepsilon}$ squares of size ε, followed by another row consisting of $\frac{W-1}{\varepsilon}$ squares of size ε followed by the square of size 1. All these rows are completely full, and only one square of size ε is left, this square is placed last. The height of the bottom-left packing is $h_{\mathrm{BL}} = 2$. Figure 31 depicts this bottom-left packing.

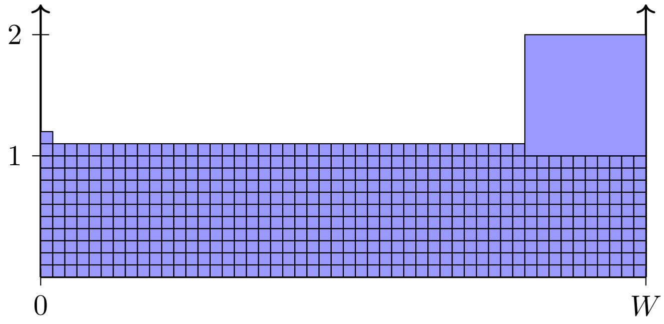


Figure 31: Bottom-left packing of instance using the LRF-ordering.

An optimum packing of the squares is depicted in Figure 32. The square of size 1 is placed on the bottom. Next to the square of size 1 there are placed $\frac{W-1}{\varepsilon^2}$ squares of size ε. This means that there are

$$\frac{W}{\varepsilon^2} + \frac{W-1}{\varepsilon} + 1 - \frac{W-1}{\varepsilon^2} = \frac{1}{\varepsilon^2} + \frac{W-1}{\varepsilon} + 1$$

left to be placed. These squares fit into at most $\lceil \frac{1}{W\varepsilon} \rceil + 1$ rows of height ε. Hence the optimum packing has height $h_{\mathrm{OPT}} \leq 1 + \frac{1}{W} + 2\varepsilon$.

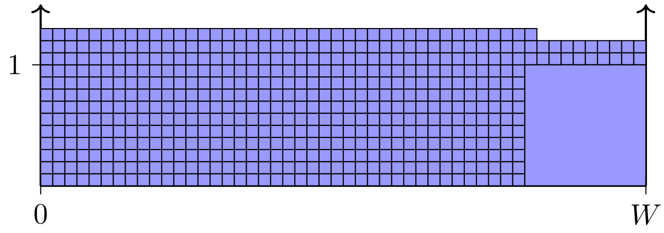


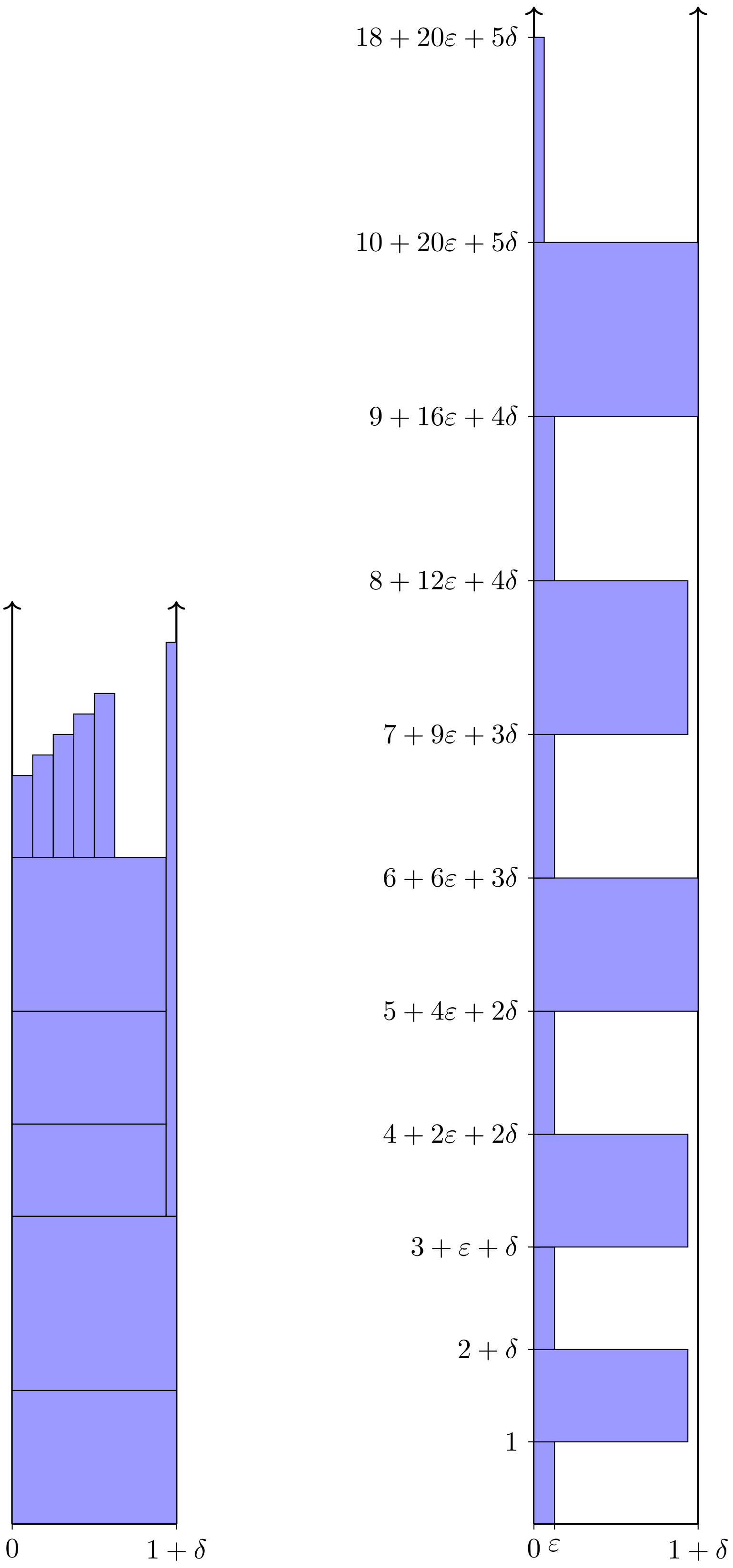
Figure 32: Optimum packing of instance.

It follows that the approximation ratio for the bottom-left algorithm of constructed instance ordered by LRF is at least

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}} \geq \frac{2}{1 + \frac{1}{W} + 2\varepsilon} = \frac{2}{1 + \frac{1}{k} + 2\varepsilon}.$$

Clearly, as $k \to \infty$ and $\varepsilon \to 0$, the approximation ratio tends to 2. ☐

Remark 5.20. As a final remark, Baker et al. [BCR80] showed that when rectangles are ordered by increasing width, then the approximation ratio of the bottom-left algorithm can be arbitrarily bad. Hence a generalization of the increasing size ordering (or last-row-full ordering) to rectangles should consider rectangles ordered by increasing height.



(a) (b)

Figure 33: The constructed instance for $n = 4$. Figure 33a shows that $h_{\mathrm{OPT}} \leq n + \mathcal{O}(\varepsilon)$. Figure 33b depicts the bottom-left packing ordered by increasing height. The top rectangle is depicted shorter for the sake of saving space. Furthermore, it shows that $h_{\mathrm{BL}}^{\text{incr-h}} = 3n + \mathcal{O}(\varepsilon)$.

However, this will probably not improve the upper bound of 3 for the bottom-left algorithm using decreasing widths, because the following instance, similar to the one in Theorem 4.1, has approximation ratio equal to 3. Namely, consider the instance consisting of the rectangles $(\varepsilon, 1 + i\varepsilon)$ for $i \in \{0, \ldots, n\}$, $(1, 1 + i\varepsilon + \delta)$ for $i \in \{0, \ldots, n\} \setminus \{\frac{n}{2}, n\}$, $(1 + \delta, 1 + i\varepsilon + \delta)$ for $i \in \{\frac{n}{2}, n\}$ and $(\delta, n + 3)$, where $\delta < \varepsilon$. Let the width of the strip be equal to $W = 1 + \delta$. The bottom-left packing of this instance ordered by increasing height is depicted in Figure 33b.

The height of the bottom-left packing is $h_{\text{BL}}^{\text{incr-h}} = 3n + \mathcal{O}(\varepsilon)$ and the height of an optimum packing is at most $h_{\text{OPT}} \leq n + \mathcal{O}(\varepsilon)$. Hence, this instance shows that the absolute approximation ratio is at least 3 for sufficiently small values of δ and ε. However, it remains an open question whether ordering rectangles by increasing height gives an asymptotic approximation ratio depending on k and p such that for many cases the approximation ratio of the bottom-left algorithm becomes better-than-3.

6 Conclusion

The main objective of this thesis was to narrow the gap between the lower and upper bound of the approximation ratio of the bottom-left algorithm. For the general `Strip Packing` case, Theorem 3.1 improved the lower bound from 5/4 to $4/(3+\varepsilon)$. Similarly, for `Square Strip Packing`, Theorem 3.3 improved the lower bound from $12/(11+\varepsilon)$ to $4/(3+\varepsilon)$.

On the other side, Theorem 4.36 showed that the approximation ratio of the bottom-left algorithm for SSPP-instances is bounded by 16 regardless of the order of the squares. This strongly contrasts the general case where the approximation ratio can be unbounded for badly ordered SPP-instances.

Ordering squares by decreasing size results in an approximation ratio of the bottom-left algorithm equal to 2 as shown by Baker et al. [BCR80]. In Theorem 5.8 and 5.12 the opposite ordering was studied, namely, ordering squares by increasing size. The vertical cover theorem (Theorem 5.7) resulted in the asymptotic approximation ratio

$$h_{\text{BL}}^{\text{incr}} \leq \frac{k}{k-1} h_{\text{OPT}} + \frac{2k-3}{k-1} h_{\max} = \frac{(p+2)k-3}{p(k-1)} h_{\text{OPT}}$$

where $k = W/h_{\max}$ and $p = h_{\text{OPT}}/h_{\max}$. The horizontal cover theorem (Theorem 5.11) resulted in the asymptotic approximation ratio

$$h_{\text{BL}}^{\text{incr}} \leq \frac{k}{k-2} h_{\text{OPT}} + \frac{k-3}{k-2} h_{\max} = \frac{(p+1)k-3}{p(k-2)} h_{\text{OPT}}.$$

These two results together imply that

$$\frac{h_{\text{BL}}^{\text{incr}}}{h_{\text{OPT}}} \leq \min\left\{ \frac{(p+2)k-3}{p(k-1)}, \frac{(p+1)k-3}{p(k-2)} \right\}.$$

This is illustrated in Table 2 for different values of k and p. For $p = 1$ the approximation ratio converges to 2 as k tends to infinity. This is tight as shown in Example 5.13. Furthermore, for $p > 1$, there exists k such that the approximation ratio becomes better-than-2, which is not the case when squares are ordered by decreasing size as shown by the checkerboard example from Baker et al. [BCR80] (Example 4.35). In other words, for $p > 1$ and large k, the approximation ratio of the bottom-left algorithm using the increasing size order is better than using the decreasing size order.

p	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$
1	3	3	2.5	2.33	2.25	2.2	2.17	2.14
1.25	2.8	2.7	2.4	2.2	2.1	2.04	2	1.97
1.5	2.67	2.5	2.33	2.11	2	1.93	1.89	1.86
1.75	2.57	2.36	2.29	2.05	1.93	1.86	1.81	1.78
2	2.5	2.25	2.17	2	1.88	1.8	1.75	1.71
2.5	2.4	2.1	2	1.93	1.8	1.72	1.67	1.63
3	2.33	2	1.89	1.83	1.75	1.67	1.61	1.57
4	2.25	1.88	1.75	1.69	1.65	1.6	1.54	1.5
5	2.2	1.8	1.67	1.6	1.56	1.53	1.5	1.46
6	2.17	1.75	1.61	1.54	1.5	1.47	1.45	1.43

Table 2: An upper bound on the approximation ratio of the bottom-left algorithm for SSPP-instances ordered by increasing size. Here $k = W/h_{\max}$ and $p = h_{\text{OPT}}/h_{\max}$.

Still, there is even a better ordering of the squares, namely the last-row-full ordering. Theorem 5.16 shows that for each SSPP-instance, the LRF ordering gives a better height than the increasing size ordering. Moreover, Theorem 5.18 proves that the bottom-left algorithm using the LRF ordering on SSPP-instances results in the asymptotic approximation ratio

$$h_{\mathrm{BL}}^{\mathrm{LRF}} \leq \frac{k}{k-1} h_{\mathrm{OPT}} + \frac{k-2}{k-1} h_{\max} = \frac{(p+1)k-2}{p(k-1)} h_{\mathrm{OPT}}.$$

It holds that for each $p > 1$ and $k > 2$, the approximation ratio of the bottom-left algorithm using the LRF ordering is better-than-2. Thus, in most cases it is better to use the LRF-ordering rather than the decreasing size ordering. And in all cases it is better to use the LRF-ordering rather than the increasing size ordering. Table 3 illustrates the approximation ratio of the bottom-left algorithm using the LRF-ordering for different values of p and k.

p	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$
1	2	2	2	2	2	2	2	2
1.25	2	1.9	1.87	1.85	1.84	1.83	1.83	1.83
1.5	2	1.83	1.78	1.75	1.73	1.72	1.71	1.71
1.75	2	1.79	1.71	1.68	1.66	1.64	1.63	1.63
2	2	1.75	1.67	1.63	1.6	1.58	1.57	1.56
2.5	2	1.7	1.6	1.55	1.52	1.5	1.49	1.48
3	2	1.67	1.56	1.5	1.47	1.44	1.43	1.42
4	2	1.63	1.5	1.44	1.4	1.38	1.36	1.34
5	2	1.6	1.47	1.4	1.36	1.33	1.31	1.3
6	2	1.58	1.44	1.38	1.33	1.31	1.29	1.27

Table 3: An upper bound on the approximation ratio of the bottom-left algorithm for SSPP-instances using the last-row-full ordering.

All in all, the research in this thesis narrowed the gap between the lower and upper bounds for the bottom-left algorithm. The lower bound got improved to $4/(3+\varepsilon)$ and the upper bound got better-than-2 for many special cases consisting of squares.

6.1 Open questions

Some open questions remain for further research. For instance, it is probably possible to improve the analysis of Chapter 4 to obtain a better-than-16 approximation ratio for the worst-case bottom-left packing of a SSPP-instance. There are three possible ways to improve this. First of all, it might be possible to better count the number of times a square is used in the global covering theorem (Theorem 4.32). Secondly, it might be possible to find better coverings of the trenches instead of cutting off part of the top of the packing. And thirdly, it might be possible to find a totally new covering of the unoccupied space that is more efficient. Additionally, it might be possible to improve the $6 - 2\sqrt{2}$ lower bound from Theorem 4.36.

Open question 1. Find an upper bound for the worst-case bottom-left packing of SSPP-instances that is smaller than 16. Or improve the $6 - 2\sqrt{2}$ lower bound.

Another question is if it is possible to generalize the ideas of Chapter 5 to instances with rectangles. As Remark 5.20 explained, it might be possible to order rectangles by increasing height and obtain an asymptotic approximation ratio that is better-than-3 in many cases.

Open question 2. Does ordering rectangles by increasing height result in an asymptotic approximation ratio depending on $k = W/h_{\max}$ and $p = h_{\mathrm{OPT}}/h_{\max}$ for the bottom-left algorithm?

Bibliography

[Kar72] Richard M. Karp. “Reducibility among combinatorial problems”. In: *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972).* Plenum, New York, 1972, pp 85–103.

[GJ78] M. R. Garey and D. S. Johnson. ““Strong” NP-completeness results: motivation, examples, and implications”. In: *J. Assoc. Comput. Mach.* 25.3 (1978), pp. 499–508. ISSN: 0004-5411,1557-735X. DOI: `10.1145/322077.322090`. URL: `https://doi.org/10.1145/322077.322090`.

[BCR80] Brenda Baker, Ed Coffman and Ronald Rivest. “Orthogonal Packings in Two Dimensions”. In: *SIAM J. Comput.* 9 (Nov. 1980), pp. 846–855. DOI: `10.1137/0209064`.

[Bro80] Donna J. Brown. “An improved BL lower bound”. In: *Information Processing Letters* 11.1 (1980), pp. 37–39. ISSN: 0020-0190. DOI: `https://doi.org/10.1016/0020-0190(80)90031-9`. URL: `https://www.sciencedirect.com/science/article/pii/0020019080900319`.

[Cha83] Chazelle. “The Bottomn-Left Bin-Packing Heuristic: An Efficient Implementation”. In: *IEEE Transactions on Computers* C-32.8 (1983), pp. 697–707. DOI: `10.1109/TC.1983.1676307`.

[Leu+90] Joseph Y.-T. Leung et al. “Packing squares into a square”. In: *J. Parallel Distrib. Comput.* 10.3 (1990), pp. 271–275. ISSN: 0743-7315. DOI: `10.1016/0743-7315(90)90019-L`. URL: `https://doi.org/10.1016/0743-7315(90)90019-L`.

[Har+14] Rolf Harren et al. “A $(5/3+\epsilon)$-approximation for strip packing”. In: *Computational Geometry* 47.2, Part B (2014). The 12th International Symposium on Algorithms and Data Structures (WADS2011), pp. 248–267. ISSN: 0925-7721. DOI: `https://doi.org/10.1016/j.comgeo.2013.08.008`. URL: `https://www.sciencedirect.com/science/article/pii/S0925772113001016`.

[KV18] Bernhard Korte and Jens Vygen. *Combinatorial optimization.* Sixth. Vol. 21. Algorithms and Combinatorics. Theory and algorithms. Springer, Berlin, 2018, pp. xxi+698. ISBN: 978-3-662-56038-9; 978-3-662-56039-6. DOI: `10.1007/978-3-662-56039-6`. URL: `https://doi.org/10.1007/978-3-662-56039-6`.