TWO-DIMENSIONAL PACKING PROBLEMS

ROLF HARREN

Dissertation

zur Erlangung des Grades des Doktors der Naturwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Saarbrücken, Juli 2010

TAG DES KOLLOQUIUMS
15. Oktober 2010

DEKAN DER FAKULTÄT 6
Prof. Dr. Holger Hermanns

MITGLIEDER DES PRÜFUNGSAUSSCHUSSES
Prof. Dr. Gerhard Weikum (Vorsitzender)
PD Dr. Rob van Stee
Prof. Dr. Kurt Mehlhorn
Prof. Dr. Klaus Jansen
Dr. Eric Berberich

ORT
Saarbrücken

Rolf Harren: *Two-dimensional packing problems,* Dissertation, Juli 2010

Für Annalena

ABSTRACT

In this thesis we consider the two-dimensional bin packing problem and the strip packing problem, which are popular geometric generalizations of the classical bin packing problem.

In both problems, a list of rectangles has to be packed into a designated area such that no two rectangles overlap and all rectangles are packed axis-parallel. For the strip packing problem, the given items have to be packed into a strip of unit width and minimal height, whereas in the two-dimensional bin packing problem a packing has to be found into a minimal number of unit-sized bins.

We investigate approximation algorithms and online algorithms for these problems and consider variants where rotations of the rectangles are forbidden and where rotations by 90 degrees are allowed. In particular, we present two approximation algorithms for strip packing with approximation ratios 1.9396 and arbitrarily close to 5/3, respectively. These results are the first improvements upon the approximation ratio of 2 that was established 16 years ago. Moreover, we show an improved lower bound of 2.589 on the competitive ratio of online strip packing along with an upper bound of 2.618 for restricted input instances. For two-dimensional bin packing we derive best-possible approximation algorithms for the variants with and without rotations.

ZUSAMMENFASSUNG

In dieser Arbeit befassen wir uns mit dem zweidimensionalen Bin Packing Problem und dem Strip Packing Problem. Beide Probleme sind geometrische Verallgemeinerungen des klassischen Bin Packings.

Bei beiden Problemen soll eine gegebene Liste an Rechtecken so in einen vorgegebenen Bereich gepackt werden, dass die Rechtecke sich nicht überlappen und alle Rechtecke achsenparallel gepackt sind. Beim Strip Packing soll die Packungshöhe einer Packung der Rechtecke in einen Streifen (Strip) der Breite 1 minimiert werden. Dahingegen erfolgt die Packung beim Bin Packing in eine minimale Anzahl an Quadraten (Bins) der Größe 1.

Wir untersuchen Approximationsalgorithmen und Onlinealgorithmen für diese Probleme und unterscheiden die Varianten, in denen die Rechtecke nicht rotiert werden dürfen und in denen eine Rotation um 90 Grad erlaubt ist. Für das Strip Packing Problem zeigen wir Approximationsalgorithmen mit Güten 1,9396 und beliebig nah an 5/3. Dies sind die ersten Verbesserungen der Approximationsgüte von 2 die vor 16 Jahren gezeigt wurde. Des Weiteren zeigen wir eine verbesserte untere Schranke von 2,589 für das online Strip Packing Problem und geben einen Onlinealgorithmus mit Güte 2,618 für eingeschränkte Instanzen an. Für das Bin Packing Problem präsentieren wir bestmögliche Algorithmen für die Varianten mit und ohne Rotation.

PUBLICATIONS

The results presented in this thesis have previously appeared or been submitted as parts of the following publications:

- R. Harren and R. van Stee. Absolute approximation ratios for packing rectangles into bins. *Journal of Scheduling*, 2009, doi:10.1007/s10951-009-0110-3
  (A preliminary version was published in SWAT 2008)
- R. Harren and R. van Stee. Improved absolute approximation ratios for two-dimensional packing problems. In *APPROX: Proc. 12th Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 177–189, 2009
- R. Harren, K. Jansen, L. Prädel and R. van Stee. A $(5/3 + \varepsilon)$-approximation for strip packing. *submitted*

The results on online strip packing are based on a joint work with Walter Kern. The publication of these results is in preparation.

ACKNOWLEDGMENTS

I would like to thank Xin Han for initially introducing me to the topic of multi-dimensional packing problems during a research stay at Kyoto University. I am thankful to Kazuo Iwama for his hospitality during this research stay and to Ingo Wegener for his help and encouragement.

Benjamin Doerr was the one to draw my attention to the Max-Planck-Institut für Informatik. I am thankful to him for that and for the continued mentorship during my time here. I was very lucky to be part of the MPI, benefitting from the excellent working conditions and facilities. My thanks go to Kurt Mehlhorn for offering me this great opportunity. The people in the working group made my stay here pleasant and enjoyable, especially my room mates Edda Happ and Stefan Kratsch.

The best experiences during my research arouse from collaborations with others. Therefore, I am grateful to my coauthors Florian Diedrich, Klaus Jansen, Walter Kern, Lars Prädel, Ralf Thöle, Henning Thomas, and Rob van Stee.

It is probably something of a rarity to be able to pinpoint a personal research breakthrough to a specific event. I am very thankful to the anonymous questioner in the discussion after a presentation at ETH Zürich. One particular question opened my eyes towards the applicability of the developed methods on a different problem. I thank Reto Spöhel for his invitation to ETH Zürich and the fruitful discussions.

Special thanks go to my advisor Rob van Stee whose thoughtful comments helped to improve the presentation of this dissertation.

I am deeply grateful to the support of my family, who never complained seeing me rarely throughout the last years. Hopefully I will be able to give back at least a fraction of the unconditional support that I received.

Finally, my biggest thank goes to my wife, Annalena. Your love is of utmost importance to me. Together with you I am cheerfully looking forward to the next chapters in our lives.

Rolf Harren

CONTENTS

1 INTRODUCTION

The classical bin packing problem is one of the most fundamental problems in combinatorial optimization. In this thesis we consider popular geometric generalizations of this problem, namely, the two-dimensional bin packing problem and the strip packing problem.

1.1 PACKING PROBLEMS

In the classical (one-dimensional) bin packing problem one is given a list of items $I = \{a_1, \ldots, a_n\}$ with sizes $s_i \in (0, 1]$ and an unlimited supply of unit-sized bins. The goal is to find an assignment (packing) of the items to a minimum number of bins such that the sum of item sizes in each bin does not exceed its unit capacity. The one-dimensional problem and its applications are subject to a great number of articles, see Coffman, Garey & Johnson[13] for a survey.

In this thesis we consider the two-dimensional bin packing problem and the strip packing problem. These geometric generalizations of the bin packing problem have received substantial research interest in recent years, we refer to Sections 1.3.1 and 1.4.1 in this introduction for further references.

In the two-dimensional bin packing problem, a list $I = \{r_1, \ldots, r_n\}$ of rectangles of width $w_i \leq 1$ and height $h_i \leq 1$ is given. An unlimited supply of square bins of unit size is available to pack all items from I such that no two items overlap and all items are packed axis-parallel into the bins. The goal is to minimize the number of bins used. For the strip packing problem, the given items have to be packed into a strip of unit width and minimal height.

For both problems two main variants have been studied, taking into consideration the rotation of items. Depending on the variant, rotations might be completely forbidden or rotation by 90 degrees are allowed. Note that the assumption of unit-sized bins is a restriction in the case where rotations are permitted.

1.2 APPLICATIONS

Classical bin packing has a large number of applications, reaching from industry (cutting material such us cables, lumber or paper) to computer systems (memory allocation in paged computer systems) and networking (packet routing in communication networks). Effectively, bin packing appears as a sub-problem in various other settings.

Additionally, there are several new applications for the geometric generalizations of bin packing that we consider here. Most importantly, in manufacturing settings rectangular pieces need to be cut out of some sheet of raw material, while minimizing the waste. Obviously, cutting problems and packing problems correspond to each other. Restrictions to orthogonal packing and packing with restricted or even without rotations make sense in this setting as well if we cut items out of patterned fabric and have to retain the alignment of the pattern. These cutting stock problems occur as bin packing and strip packing problems.

Scheduling independent tasks on a group of processors, each requiring a certain number of contiguous processors or memory allocation during a certain length of time, can also be modeled as a strip packing problem[37]. In this application the width of the strip represents the total number of processors or memory available, and the height represents the maximal completion time. Thinking of (semi-)manually operated machines instead of processors we might have periodic breaks, as they would occur when working in shifts, making this a two-dimensional bin packing problem.

Further applications can also be found in VLSI-design (minimum rectangle placement problem[7]) and in the advertisement placement problem[19]. In this problem, we have to place all given rectangular ads on a minimal number of pages or web-pages.

In this thesis we develop approximation algorithms, that get the complete input at once and have to compute an approximate solution in polynomial time, and we consider online algorithms, that receive the input items one after the other and have to irrevocably place each item before the next item is released. For both types of algorithms, the solution derived by the algorithm is compared to an optimal solution.

In the following sections we introduce both of these concepts along with the corresponding related work on two-dimensional packing and a short description of our contribution.

1.3 APPROXIMATION ALGORITHMS

The one-dimensional bin packing problem was one of the first problems to be proven $\mathcal{NP}$-complete[20] and, according to Garey, Coffman & Johnson[13], one of the first problems for which the idea of worst-case performance guarantees was investigated. Of course, also any generalization that contains the original problem is $\mathcal{NP}$-complete. As we cannot expect to find efficient algorithms for $\mathcal{NP}$-complete problems, research has concentrated on finding good approximation algorithms. In this thesis we use the following two notions of the approximation ratio.

Let $\text{ALG}(I)$ be the value, i.e., the height of the strip or the number of bins, of a packing produced by algorithm ALG on input I. Denote an optimal algorithm by OPT. The (absolute) approximation ratio of packing algorithm ALG is defined to be

$$\rho_{\text{ALG}} = \sup_I \left\{ \frac{\text{ALG}(I)}{\text{OPT}(I)} \right\}.$$

A large portion of the previous work on two-dimensional packing problems has focused on the *asymptotic* approximation ratio, i.e., the behavior of the algorithm on instances with large optimal value. The asymptotic approximation ratio of packing algorithm ALG is defined to be

$$\rho^{\infty}_{\text{ALG}} = \limsup_{n\to\infty} \sup_I \left\{ \frac{\text{ALG}(I)}{\text{OPT}(I)} \;\middle|\; \text{OPT}(I) = n \right\}.$$

A problem admits a polynomial-time approximation scheme ($\mathcal{PTAS}$) if there is a family of algorithms $\{A_\varepsilon \mid \varepsilon > 0\}$ such that for any $\varepsilon > 0$ and any instance I, A_ε produces a $(1+\varepsilon)$-approximate solution in time polynomial in the size of the input. A fully polynomial-time approximation scheme ($\mathcal{FPTAS}$) is a $\mathcal{PTAS}$ where additionally A_ε has run-time polynomial in $1/\varepsilon$. Asymptotic (fully) polynomial-time approximation schemes are similarly defined in terms of the asymptotic approximation ratio.

1.3.1 KNOWN RESULTS

Due to its prominence and importance, one-dimensional bin packing received a lot of research attention. The most important results include an asymptotic $\mathcal{PTAS}$ by Fernandez de la Vega & Lueker[18] and an asymptotic $\mathcal{FPTAS}$ by Karmarkar & Karp[33]. On the negative side, it is well known that bin packing cannot be approximated within a factor of $3/2 - \varepsilon$ for any $\varepsilon > 0$. We also refer to the survey of Coffman, Garey & Johnson[13] for more details on this problem. In the following, we present in more detail the results on the two-dimensional problems that we investigate in this thesis.

STRIP PACKING. Coffman, Garey, Johnson & Tarjan[14] studied algorithms based on methods for one-dimensional bin packing. Their algorithms, which pack the rectangles on shelves, are called NEXT FIT DECREASING HEIGHT (NFDH) and FIRST FIT DECREASING HEIGHT (FFDH). They proved that $\text{NFDH}(I) \leq 2\,\text{OPT}(I) + h_{\max}(I)$ and $\text{FFDH}(I) \leq 1.7\,\text{OPT}(I) + h_{\max}(I)$, where $h_{\max}(I)$ is the maximal height among all rectangles in the instance I. The approximation algorithm presented by Sleator[42] generates a packing of height $2\,\text{OPT}(I) + h_{\max}(I)/2$.

The asymptotic performance ratio of the above algorithms was further reduced to $4/3$ by Golan[21] and then to $5/4$ by Baker, Brown & Katseff[1]. These algorithms generate packings of heights at most $4/3 \cdot \text{OPT}(I) + 127/18 \cdot h_{\max}(I)$ and $5/4 \cdot \text{OPT}(I) + 53/8 \cdot h_{\max}(I)$, respectively.

Kenyon & Rémila[34] and Jansen & van Stee[29] gave asymptotic $\mathcal{FPTAS}$'s for strip packing without rotations and with rotations, respectively. The additive constant was recently improved from $\mathcal{O}(1/\varepsilon^2)$ to $h_{\max}(I)$ by Jansen & Solis-Oba[27] at the cost of a higher running time, i.e., losing the polynomial runtime bound in $1/\varepsilon$ and making the algorithm an asymptotic $\mathcal{PTAS}$.

In terms of absolute approximability, Baker, Coffman & Rivest[2] showed that the BOTTOM LEFT algorithm has an approximation ratio equal to 3 when the rectangles are ordered by decreasing widths. Many of the results above also imply an absolute performance guarantee. The results of Coffman et al.[14] and of Sleator[42] directly imply absolute approximation ratios of 3, 2.7 and 2.5 for their algorithms, respectively, since $h_{\max}(I) \leq \text{OPT}(I)$.

This was independently improved by Schiermeyer[39] and Steinberg[43] with algorithms of approximation ratio 2. Especially Steinberg's algorithm has been used in many subsequent bin packing and strip packing papers as a subroutine. Jansen & Solis-Oba[27] showed an absolute $\mathcal{PTAS}$ for strip packing with rotations on instances with optimal height at least 1. After the work by Steinberg and Schiermeyer in 1994, there was no improvement on the best known absolute approximation ratio when rotations are forbidden until very recently. Jansen & Thöle[28] presented an approximation algorithm with approximation ratio arbitrarily close to $3/2$ for restricted instances where the widths are of the form i/m for $i \in \{1, \ldots, m\}$ and m is polynomially bounded in the number of items. Notice that the general version that we consider appears to be considerably more difficult.

On the negative side, since strip packing includes the one-dimensional bin packing problem as a special case, there is no algorithm with absolute ratio better than $3/2$ unless $\mathcal{P} = \mathcal{NP}$.

TWO-DIMENSIONAL BIN PACKING. In 1982, Chung, Garey & Johnson[12] proposed the algorithm HYBRID FIRST FIT for two-dimensional bin packing and proved that its asymptotic approximation ratio is at most 2.125. Caprara[9] was the first to present an algorithm with an asymptotic approximation ratio less than 2 for two-dimensional bin packing. Indeed, he considered 2-stage packing, in which the items must first be packed into shelves that are then packed into bins, and showed that the asymptotic worst case ratio between two-dimensional bin packing and 2-stage packing is $T_\infty = 1.691\ldots$. Therefore, the asymptotic $\mathcal{FPTAS}$ for 2-stage packing by Caprara, Lodi & Monaci[10] achieves an asymptotic approximation guarantee arbitrarily close to T_∞.

In 2006, Bansal, Caprara & Sviridenko[6] presented a general framework to improve subset oblivious algorithms and obtained asymptotic approximation guarantees arbitrarily close to $1.525\ldots$ for packing with rotations of 90 degrees or without rotations. These are the currently best-known asymptotic approximation ratios for general two-dimensional bin packing problems. For packing squares into square bins, Bansal, Correa, Kenyon & Sviridenko[7] gave an asymptotic $\mathcal{PTAS}$. On the other hand, the same paper showed the $\mathcal{APX}$-hardness of two-dimensional bin packing without rotations, thus no asymptotic $\mathcal{PTAS}$ exists unless $\mathcal{P} = \mathcal{NP}$. Chlebík & Chlebíková[11] were the first to give explicit lower bounds of $1 + 1/3792$ and $1 + 1/2196$ on the asymptotic approximability of rectangle packing with and without rotations, respectively.

Not much has been done in terms of absolute approximability of the bin packing problem prior to our work. Zhang[47] presented an absolute 3-approximation algorithm. For the special case of packing squares into bins, van Stee[44] showed that an absolute 2-approximation is possible. On the negative side, Leung et al.[36] showed that it is strongly $\mathcal{NP}$-complete to decide whether a set of *squares* can be packed into a given square. Thus it is not possible to approximate two-dimensional bin packing within a factor of less than 2 unless $\mathcal{P} = \mathcal{NP}$ even if rotations by 90 degrees are allowed.

It should be noted that in the asymptotic setting, especially for the positive results for bin packing mentioned above, the approximation ratio only gets close to the stated value for very large inputs. In particular, the 1.525-approximation by Bansal et al.[6] has an additive constant which is not made explicit in the paper but which the authors admit is extremely large[4]. Thus, for any reasonable input, the actual (absolute) approximation ratio of their algorithm is much larger than 1.525, and it therefore makes sense to consider alternative algorithms under the absolute performance measure. Proving a bound on the absolute approximation gives us a performance guarantee for all inputs, not just for (very) large ones.

1.3.2 OUR CONTRIBUTION

In this thesis we focus on the absolute approximation ratio. Attaining a good absolute approximation ratio is more difficult than attaining a good asymptotic approximation ratio, because in the second case an algorithm is allowed to "waste" a constant number of bins, which allows, e.g., the classification of items followed by a packing where each class is packed separately.

For the strip packing problem, we present two approximation algorithms that break the bound of 2. Although Schiermeyer[39] already expected in his work in 1994 that the approximation ratio can be reduced below 2, these algorithms represent the first improvements on the absolute approximability of strip packing since Schiermeyer's work. First, we present an algorithm with an absolute approximation ratio of 1.9396 and second, partially based on this result but involving a much deeper investigation in different cases of packings, we present an algorithm with an absolute approximation ratio arbitrarily close to $5/3$.

These results significantly reduce the gap between the lower bound of $3/2$ and the upper bound.

For the bin packing problem, we consider the variants without rotations and with rotations by 90 degrees. For both variants we give approximation algorithms with an absolute approximation ratio of 2. Due to the result by Leung et al.[36] this is optimal provided $\mathcal{P} \neq \mathcal{NP}$. Jansen, Prädel & Schwarz[26] independently achieved the same result for bin packing without rotations. Furthermore, we prove a conjecture by Zhang[47] on the absolute approximation ratio of the HYBRID FIRST FIT (HFF) algorithm by showing that this ratio is 3.

1.4 ONLINE ALGORITHMS

In the online version the rectangles are given to the online algorithm one by one from a list, and the next rectangle is given as soon as the current rectangle is irrevocably placed. To evaluate the performance of an online algorithm we employ competitive analysis. As for the approximation algorithms, we denote the value, i.e., the height of the strip or the number of bins, of a packing produced by algorithm ALG on an input list L by $\mathrm{ALG}(L)$. The optimal algorithm OPT is not restricted in any way by the ordering of the rectangles in the list. Competitive analysis measures the absolute worst-case performance of online algorithm ALG by its competitive ratio

$$\rho_{\mathrm{abs}} = \sup_L \left\{ \frac{\mathrm{ALG}(L)}{\mathrm{OPT}(L)} \right\}.$$

As for the approximation algorithms, research has partially concentrated on the asymptotic competitive ratio that is defined as

$$\rho^{\infty}_{\mathrm{ALG}} = \limsup_{n\to\infty} \sup_{I} \left\{ \frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)} \;\middle|\; \mathrm{OPT}(I) = n \right\}.$$

Note that we use the same letter ρ for the approximation ratio and the competitive ratio as both ratios are defined similarly and describe the same: How well does a given algorithm perform relative to the optimal solution on worst-case instances. The setting, i.e., whether we consider approximation or online algorithms, only restricts the algorithms (to polynomial running time and applicability to online settings, respectively) but does not affect the optimal solution. Note that although many online algorithms are time-efficient, they are, in contrast to approximation algorithms, not constrained in their running time.

1.4.1 KNOWN RESULTS

The classical online bin packing problem has been studied extensively. The most important results include the algorithm HARMONIC++ by Seiden[40] with an asymptotic competitive ratio of 1.58889 and a lower bound on the asymptotic competitive ratio of 1.54014 by van Vliet[45].

STRIP PACKING. Baker & Schwarz[3] were the first to study so-called shelf algorithms for strip packing that do not rely on the order of the items and thus apply to the online scenario. One of their algorithms, presented already in 1983, achieves an asymptotic competitive ratio arbitrarily close to 1.7. In 1997, Csirik & Woeginger[16] showed that no shelf algorithm can achieve an asymptotic competitive ratio less than $T_\infty = 1.691\ldots$ and at the same time presented an online algorithm whose asymptotic worst-case ratio comes arbitrarily close to this value.

Han, Iwama, Ye & Zhang[23] gave a general algorithmic framework between one-dimensional bin packing and strip packing, with which they achieve the same asymptotic bounds by applying bin packing algorithms to strip packing. This makes the modified HARMONIC++ algorithm by Seiden[40] to the current champignon in terms of asymptotic competitiveness with a ratio of 1.58889. The lower bound of 1.54014 by van Vliet[45] on one-dimensional bin packing of course also holds for strip packing.

The first fit shelf algorithm by Baker & Schwarz[3] was the first algorithm with a proven *absolute* competitive ratio of 6.99 under the assumption that the height of the rectangles is at most 1. Recent advances have been made by Ye, Han & Zhang[46] and Hurink & Paulus[25]. Independently they showed that a modification of the well-known shelf algorithm yields an online algorithm with competitive ratio $7/2 + \sqrt{10} \approx 6.6623$. Both results do not assume a bound on the maximal height of the rectangles.

In the early 80's, Brown, Baker & Katseff[8] derived a lower bound of 2 on the competitive ratio of any online algorithm by constructing certain adversary sequences. These sequences were further studied by Johannes[31] and Hurink & Paulus[24], who derived improved lower bounds of 2.25 and 2.43, respectively. (Both results are computer aided and presented in terms of online parallel machine scheduling.) The paper of Hurink & Paulus[24] also presents an upper bound of 2.5 for packing such "Brown-Baker-Katseff sequences". Kern & Paulus[35] finally settled the question how well the Brown-Baker-Katseff sequences can be packed by giving a matching upper and lower bound of $3/2 + \sqrt{33}/6 \approx 2.457$.

TWO-DIMENSIONAL BIN PACKING. Regarding online packing problems we focus on strip packing. Therefore, we only state the most important results on bin packing for the sake of completeness. Han, Chin, Ting & Zhang[22] gave an algorithm with the currently best known asymptotic competitive ratio of 2.5545, improving upon the algorithm with competitive ratio 2.66013 presented by Seiden & van Stee[41]. We are not aware of any analysis with specific regards to the absolute competitiveness. Coppersmith & Raghavan[15], however, showed a bound of $3.25\,\mathrm{OPT}(I) + 8$ for their algorithm, yielding an absolute competitive ratio of at most 11.25.

As for lower bounds, Seiden & van Stee[41] considered multi-dimensional square (and cube) packing and derived a lower bound of 1.62176 on the asymptotic competitive ratio for the two-dimensional problem. Their bound was improved to 1.64062 by Epstein & van Stee[17].

1.4.2 OUR CONTRIBUTION

Using modified Brown-Baker-Katseff sequences we show an improved lower bound of $2.589\ldots$ on the absolute competitive ratio of this problem. The modified sequences that we use consist solely of two types of items, namely, *thin* items that have negligible width (and thus can all be packed in parallel) and *blocking* items that have width 1.

On the positive side, we present an online algorithm for packing any sequence composed of thin and blocking items with competitive ratio $(3 + \sqrt{5})/2 = 2.618\ldots$. This upper bound is especially interesting as it not only applies to the concrete adversary instances that we use to show our lower bound.

1.5 ORGANISATION.

This thesis is organized in two parts.

In Part I we present our work on approximation algorithms. In Chapter 2 we introduce the relevant notations and methods that we use. Chapter 3 contains our results on strip packing with the 1.9396-approximation algorithm in Section 3.1 and the $(5/3 + \varepsilon)$-approximation algorithm in Section 3.2. In Chapters 4 and 5 we deal with the bin packing problem with and without rotations, respectively, and present our 2-approximation algorithms in Sections 4.1 and 5.1. In Section 5.2, we analyse the algorithm HYBRID FIRST FIT and prove that it has an absolute approximation guarantee of 3.

In Part II we present our contribution on online algorithms. This part consists solely of Chapter 6 in which we present the lower bound of $2.589\ldots$ for online strip packing and the upper bound of $(3 + \sqrt{5})/2$ for packing instances consisting of thin and blocking items.

Part I

APPROXIMATION ALGORITHMS

2 IMPORTANT TOOLS AND PREPARATIONS

Let $I = \{r_1, \ldots, r_n\}$ be the set of given rectangles, where $r_i = (w_i, h_i)$. For a given packing P we denote the bottom left corner of an item r_i by (x_i, y_i) and its top right corner by (x'_i, y'_i), where $x'_i = x_i + w_i$ and $y'_i = y_i + h_i$. So the interior of rectangle r_i covers the area $(x_i, x'_i) \times (y_i, y'_i)$ and two rectangles are non-overlapping if their interiors are disjoint. It will be clear from the context to which packing P the coordinates refer.

Let $W_\delta = \{r_i \mid w_i > \delta\}$ be the set of so-called *δ-wide* items and let $H_\delta = \{r_i \mid h_i > \delta\}$ be the set of *δ-high* items. To simplify the presentation, we refer to the $1/2$-wide items as *wide* items and to the $1/2$-high items as *high* items. Let $W = W_{1/2}$ and $H = H_{1/2}$ be the sets of wide and high items, respectively. The set of *small* items, i.e., items r_i with $w_i \leq 1/2$ and $h_i \leq 1/2$, is denoted by S. Finally, we call items that are wide and high at the same time *big* and denote the set of big items by B.

For a set T of items, let $\mathcal{A}(T) = \sum_{r_i \in T} w_i h_i$ be the total area and let $h(T) = \sum_{r_i \in T} h_i$ and $w(T) = \sum_{r_i \in T} w_i$ be the total height and total width, respectively. Furthermore, let $w_{\max}(T) = \max_{r_i \in T} w_i$ and $h_{\max}(T) = \max_{r_i \in T} h_i$. In the following tables we give a summary of the notations used.

item r_i	
size	(w_i, h_i)
bottom left corner	(x_i, y_i)
top left corner	(x'_i, y'_i)
interior	$(x_i, x'_i) \times (y_i, y'_i)$

sets of items			
wide	W	$=$	$\{r_i \mid w_i > 1/2\}$
δ-wide	W_δ	$=$	$\{r_i \mid w_i > 1 - \delta\}$
high	H	$=$	$\{r_i \mid h_i > 1/2\}$
δ-high	H_δ	$=$	$\{r_i \mid h_i > 1 - \delta\}$
small	S	$=$	$\{r_i \mid w_i, h_i \leq 1/2\}$
big	B	$=$	$\{r_i \mid w_i, h_i > 1/2\}$

area, height and width of a set T of items			
total area	$\mathcal{A}(T)$	$=$	$\sum_{r_i \in T} w_i h_i$
total height	$h(T)$	$=$	$\sum_{r_i \in T} h_i$
total width	$w(T)$	$=$	$\sum_{r_i \in T} w_i$
maximal height	$h_{\max}(T)$	$=$	$\max_{r_i \in T} h_i$
maximal width	$w_{\max}(T)$	$=$	$\max_{r_i \in T} w_i$

We now present important subroutines of our algorithms, namely Steinberg's algorithm[43], the Next Fit Decreasing Height algorithm and an algorithm by Bansal, Caprara, Jansen, Prädel & Sviridenko[5]. Moreover, we prove the existence of a structured packing of certain sets of wide and high items.

STEINBERG'S ALGORITHM. Steinberg[43] proved the following theorem for his algorithm that we use as a subroutine multiple times.

Theorem 1 (Steinberg's algorithm[43]). *If the following inequalities hold,*

$$w_{\max}(T) \le a, \quad h_{\max}(T) \le b, \text{ and}$$
$$2\mathcal{A}(T) \le ab - (2w_{\max}(T) - a)_+(2h_{\max}(T) - b)_+$$

where $x_+ = \max(x, 0)$, then it is possible to pack all items from T into $R = (a, b)$ in time $\mathcal{O}((n \log^2 n)/\log\log n)$.

In our algorithms, we will repeatedly use the following corollary of this theorem.

Corollary 1 (Jansen & Zhang[30]). *If the total area of a set T of items is at most $1/2$ and there are no wide items (except a possible big item) then the items in T can be packed into a bin of unit size in time $\mathcal{O}((n \log^2 n)/\log\log n)$.*

Obviously, this corollary also holds for the case that there are no high items (except a possible big item). This corollary is an improvement upon Theorem 1 if there is a big item in T as in this case Theorem 1 would give a worse area bound.

NEXT FIT DECREASING HEIGHT. The NEXT FIT DECREASING HEIGHT (NFDH) algorithm was introduced for squares by Meir & Moser[38] and generalized to rectangles by Coffman, Garey, Johnson & Tarjan[14]. It is given as follows. Sort the items by non-increasing order of height and pack them one by one into layers. The height of a layer is defined by its first item, further items are added bottom-aligned with the first item and left-aligned with the previously packed item until an item does not fit. In this case this item opens a new layer. The algorithm stops if it runs out of items or a new layer does not fit into the designated area. The running time of the algorithm is $\mathcal{O}(n \log n)$. The following lemma is an easy generalization of the result from Meir & Moser.

Lemma 1. *If a given set T of items is packed into a rectangle $R = (a, b)$ by* NFDH, *then either a total area of at least*

$$(a - w_{\max}(T)) \cdot (b - h_{\max}(T))$$

is packed or the algorithm runs out of items, i.e., all items are packed.

KNAPSACK PACKING. Bansal, Caprara, Jansen, Prädel & Sviridenko[5] considered the two-dimensional knapsack problem in which each item $r_i \in I$ has an associated profit p_i and the goal is to maximize the total profit that is packed into a unit-sized bin. Using a very technical *Structural Lemma* they showed the following theorem.

Theorem 2 (Bansal, Caprara, Jansen, Prädel & Sviridenko[5]). *For any fixed $r \ge 1$ and $\delta > 0$, there exists an algorithm that returns a packing of value at least*

$$(1 - \delta)\mathrm{OPT}_{2-\mathrm{KP}}(I)$$

for instances I for which $p_i/\mathcal{A}(r_i) \in [1, r]$ for $r_i \in I$. The running time of the algorithm is polynomial in the number of items.

Here $\mathrm{OPT}_{2-\mathrm{KP}}(I)$ denotes the maximal profit that can be packed in a bin of unit size. In the case that $p_i = w_i h_i$ for all items $r_i \in I$ we want to maximize the total packed area. Let $\mathrm{OPT}_{(a,b)}(T)$ denote the maximum area of items from T that can be packed into the rectangle (a, b), where individual items in T do not necessarily fit in (a, b). By appropriately scaling the bin, the items and the accuracy we get the following corollary.

Corollary 2. *For any fixed $\delta > 0$, the $\mathcal{PTAS}$ from [5] returns a packing of $I' \subseteq I$ in a rectangle of width $a \leq 1$ and height $b \leq 1$ such that*

$$\mathcal{A}(I') \geq (1 - \delta)\mathrm{OPT}_{(a,b)}(I).$$

EXISTENCE OF STRUCTURED PACKINGS. We show that for any set of wide and high items that fits into a strip of height 1, there exists a packing of the high items and of wide items with at least half of their total height with a nice structure, i.e., such that the wide and the high items are packed in stacks in different corners of the strip. We will later see that such a packing can be approximated arbitrarily well.

Lemma 2. *For sets $H' \subseteq H$ and $W' \subseteq W \setminus H'$ of high and wide items with $\mathrm{OPT}(W \cup H) \leq 1$ there exists a packing of $W^* \cup H'$ with $W^* \subseteq W'$ and $h(W^*) \geq h(W')/2$ such that the high items are stacked in the top left and the items from W^* are stacked in the bottom right corner of the strip.*

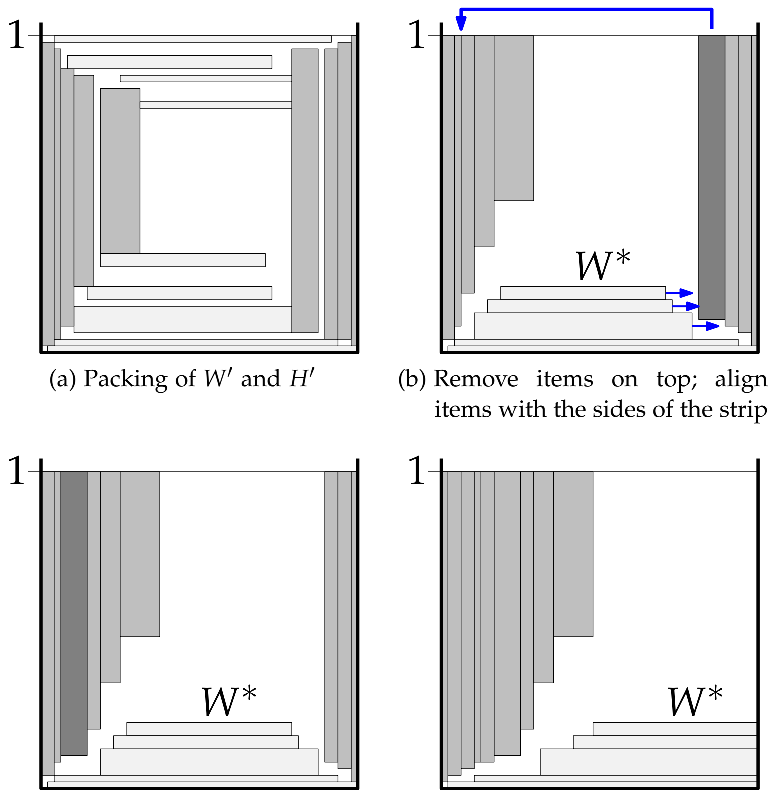
Proof. See Figure 1 for an illustration of the following proof. Consider a packing of high items H' and wide items W' into a strip of height 1. Associate each wide item with the closer boundary of the packing, i.e., either the top or bottom of the strip (an item that has the same distance to both sides of the strip can be associated with an arbitrary side). Assume w.l.o.g. that the total height of the items associated with the bottom is at least as large as the total height of the items associated with the top of the strip. Remove the items that are associated with the top and denote the other wide items by W^*. Push the items of W^* together into a stack that is aligned with the bottom of the strip by moving them purely vertically and move the high items such that they are aligned with the top of the strip and form stacks at the left and right side of the strip. Order the stacks of the high items by non-increasing order of height and the stack of the wide items by non-increasing order of width.

Now apply the following process. Take the shortest item with respect to the height from the right stack of the high items and insert it at the correct position into the left stack, i.e., such that the stack remains in the order of non-increasing heights. Move the wide items to the right if this insertion causes an overlap. Obviously this process moves all high items to the left and retains a feasible packing. In the end, all high items form a stack in the top left corner of the strip. Move the wide items to the right such that they form a stack in the bottom right corner of the strip. □

We constructively use the previous existence result with the following Corollary.

Corollary 3. *For sets $H' \subseteq H$ and $W' \subseteq W \setminus H'$ of high and wide items with $\mathrm{OPT}(W \cup H) \leq 1$ we can derive a packing of $W' \cup H'$ into a strip of height at most*

$$1 + \frac{h(W')}{2}$$



(a) Packing of W' and H'

(b) Remove items on top; align items with the sides of the strip

(c) Insert the shortest item from the right in the stack on the left; move W^* horizontally

(d) Final packing with stacks in the bottom right and top left

Figure 1: The insertion process of Lemma 2

such that the wide items are stacked in the bottom right of the strip and the high items are stacked above the wide items at the left side of the strip in time $\mathcal{O}(n \log n)$.

Proof. Consider a packing of height 1 of $W^* \cup H'$ with $W^* \subseteq W'$ and $h(W^*) \geq h(W')/2$ such that the wide items from W^* are stacked in the bottom right corner of the strip and the high items are stacked above W^* at the top left of the strip. Such a packing exists by Lemma 2. Now move up $W^* \cup H'$ by $h(W' \setminus W^*)$, pack $W' \setminus W^*$ below W^* and restore the order in the stack of the wide items. This does not cause a conflict as the original outline of W^* is not violated. As $h(W' \setminus W^*) = h(W') - h(W^*) \leq h(W')/2$ the height bound of the corollary in satisfied. Since the packing only consists of the ordered stacks of the high and wide items, we can easily derive a packing of at most the same height in time $\mathcal{O}(n \log n)$ by building the stacks and moving down the stack of H' as far as possible. $\square$

3 STRIP PACKING

In this chapter we present two algorithms for strip packing, namely, a 1.9396-approximation algorithm in Section 3.1 and an algorithm that achieves an approximation ratio arbitrarily close to 5/3 in Section 3.2. The methods in Section 3.1 are crucial not only for the following section on strip packing but also for our 2-approximation for bin packing without rotation in Section 5.1.

An important link between strip packing and two-dimensional bin packing is the interpretation of a strip of height 1 as a bin of unit size. This link is especially crucial as handling instances that fit into one bin turns out to be a major challenge for bin packing. Moreover, strip packing can essentially be reduced to the packing of instances with optimal value at most 1 as the following lemma shows.

Lemma 3. *If there exists a polynomial-time algorithm for strip packing that packs any instance I with optimal value at most 1 into a strip of height $h \geq 1$, then there also exists a polynomial-time algorithm for strip packing with absolute approximation ratio at most $h + \varepsilon$.*

Proof. Let ALG be the algorithm that packs any instance I with optimal value at most 1 into a strip of height h and assume that $h \leq 2$ by otherwise applying Steinberg's algorithm. Let ε' be the maximal value with $\varepsilon' \leq \varepsilon/(2h)$ such that $1/\varepsilon'$ is integer. We guess the optimal value approximately and apply ALG on an appropriately scaled instance. To do this, we first apply Steinberg's algorithm on I to get a packing into height $h' \leq 2\,\mathrm{OPT}(I)$. We split the interval $J = [h'/2, h']$ into $1/\varepsilon'$ subintervals

$$J_i = \left[\frac{(1+\varepsilon'(i-1))h'}{2}, \frac{(1+\varepsilon' i)h'}{2}\right]$$

for $i = 1, \dots, 1/\varepsilon'$. Then we iterate over $i = 1, \dots, 1/\varepsilon'$, scale the heights of all items by $2/((1+\varepsilon' i)h')$ and apply the algorithm ALG on the scaled instance I'. Convert the packing to a packing of the unscaled instance I and finally output the minimal packing that was derived. We eventually consider $i^* \in \{1, \dots, 1/\varepsilon'\}$ with $\mathrm{OPT}(I) \in J_{i^*}$. Then we have

$$1 - \varepsilon' < 1 - \frac{\varepsilon'}{1+\varepsilon' i^*} = \frac{(1+\varepsilon'(i^*-1))\frac{h'}{2}}{(1+\varepsilon' i^*)\frac{h'}{2}} \leq \mathrm{OPT}(I') \leq \frac{(1+\varepsilon' i^*)\frac{h'}{2}}{(1+\varepsilon' i^*)\frac{h'}{2}} = 1$$

and thus

$$\frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)} = \frac{\mathrm{ALG}(I')}{\mathrm{OPT}(I')} < \frac{h}{1-\varepsilon'} = h + \frac{\varepsilon' h}{1-\varepsilon'} \leq h + 2\varepsilon' h \leq h + \varepsilon. \qquad \square$$

Thus we concentrate on approximating instances that fit into a strip of height 1 and therefore assume $\mathrm{OPT}(I) \leq 1$ for the remainder of this chapter.

3.1 A 1.9396-APPROXIMATION ALGORITHM FOR STRIP PACKING

The overall approach for our 1.9396-approximation algorithm consists of two steps. First, we use the $\mathcal{PTAS}$ from [5] to pack instances where the total height of the $(1-\delta)$-wide items is small relative to δ into a strip of height $2-x$ for some positive value x and some $\delta \in (0, 1/2)$. Second, we derive an area guarantee for instances that could not be packed in the previous step and use this guarantee to successfully pack the instance into a strip of height $2-x$.

Finally, we will show that x can be chosen as large as $(1-\ln 2)/(3+3\ln 2) - \varepsilon$ and with Lemma 3 we get the following theorem for any $0 < \varepsilon < 10^{-5}/2$.

Theorem 3. *There exists a polynomial-time approximation algorithm for strip packing with absolute approximation ratio*

$$2 - x + \varepsilon = \frac{5 + 7\ln 2}{3 + 3\ln 2} + 2\varepsilon < 1.9396.$$

In the following assume that we have a fixed $x \in [0, 1/6 - 5/3\,\varepsilon)$ and $0 < \varepsilon \leq 10^{-5}/2$.

3.1.1 SMALL TOTAL HEIGHT OF THE $(1-\delta)$-WIDE ITEMS

We now describe an important subroutine that is used by our algorithms for strip and bin packing. Consider the case that the total height of the $(1-\delta)$-wide items is small relative to δ, i.e.,

$$h(W_{1-\delta}) \leq \frac{\delta(1-x) - 2x - \varepsilon}{1 + 2\delta} =: f(\delta)$$

for some $\delta \in ((2x+\varepsilon)/(1-x), 1/2]$ (the lower bound is required as otherwise $f(\delta) < 0$). We want to derive a packing of I into two bins such that only a height of $1-x$ is used in the second bin. For strip packing this directly gives a height of $2-x$ by putting the second bin on top of the first. And for bin packing we get a feasible solution for all $x \geq 0$, which we use in Section 5.1.

Let

$$\gamma := f(\delta) + x = \frac{\delta(1+x) - x - \varepsilon}{1 + 2\delta} < \frac{1}{2}.$$

In the first step, we show that a packing of almost all items into a unit bin and with a special structure exists. This special structure consists of a part of width $w(H_{1-\gamma})$ for the $(1-\gamma)$-high items and a part of width $1 - w(H_{1-\gamma})$ for the other items. The following lemma shows that almost all other items can be packed. Recall that $\mathrm{OPT}_{(a,b)}(T)$ denotes the maximum area of items from T that can be packed into the rectangle (a, b) as introduced for Corollary 2.

Lemma 4. *We have* $\mathrm{OPT}_{(1-w(H_{1-\gamma}),1)}(I \setminus H_{1-\gamma}) \geq \mathcal{A}(I \setminus H_{1-\gamma}) - 2\gamma$.

Proof. Consider an optimal packing of I into a bin. Remove all items that are completely contained in the top or bottom γ-margin. After this step there is no item directly above or below any item of $H_{1-\gamma} = \{r_i \mid h_i > 1-\gamma\}$. Thus we can cut the remaining packing at the left and right side of any item from $H_{1-\gamma}$. These cuts partition the packing into parts which can be swapped without losing any further items. Move all items of $H_{1-\gamma}$ to the left of the bin and move all other parts of the packing to the right. The total area of the removed items is at most 2γ and thus a total area of at least $\mathcal{A}(I \setminus H_{1-\gamma}) - 2\gamma$ fits into the rectangle of size $(1 - w(H_{1-\gamma}), 1)$ to the right of $H_{1-\gamma}$. □

In the second step, we actually derive a feasible packing that is based on the structure described above (see Figure 2). First, pack $H_{1-\gamma}$ into a stack of width $w(H_{1-\gamma})$ at the left side of the first bin. Note that $w(H_{1-\gamma}) \leq 1$ since $\gamma < 1/2$. This leaves an empty space of width $1 - w(H_{1-\gamma})$ and height 1 at the right. We therefore apply the $\mathcal{PTAS}$ from [5] on $I \setminus H_{1-\gamma}$ and a rectangle of size $(1 - w(H_{1-\gamma}), 1)$ using an accuracy of ε. Lemma 4 and Corollary 2 yield that at least a total area of $\mathcal{A}(I \setminus H_{1-\gamma}) - 2\gamma - \varepsilon$ is packed by the algorithm.

Let T be the set of remaining items. We have $\mathcal{A}(T) \leq 2\gamma + \varepsilon$. Pack the remaining $(1-\delta)$-wide items, i.e., the items of $T \cap W_{1-\delta}$, in a stack at the bottom of the second bin. The total area of the remaining items $T \setminus W_{1-\delta}$ is

$$\mathcal{A}(T \setminus W_{1-\delta}) \leq \mathcal{A}(T) - (1-\delta)h(T \cap W_{1-\delta}) \leq 2\gamma + \varepsilon - (1-\delta)h(T \cap W_{1-\delta}).$$

We pack these items with Steinberg's algorithm into the free rectangle of size (a, b) with $a = 1$ and $b = 1 - h(T \cap W_{1-\delta}) - x$ above the stack of $T \cap W_{1-\delta}$ in the second bin. We have $w_{\max}(T \setminus W_{1-\delta}) \leq 1 - \delta \leq 1$ and since $\gamma = f(\delta) + x \geq h(W_{1-\delta}) + x$ we have

$$h_{\max}(T \setminus W_{1-\delta}) \leq 1 - \gamma \leq 1 - h(W_{1-\delta}) - x \leq 1 - h(T \cap W_{1-\delta}) - x = b.$$

Steinberg's algorithm is applicable as we have $\delta < 1/2$ and $\gamma < 1/2$ and thus we get

$$\begin{aligned}
& 2\mathcal{A}(T \setminus W_{1-\delta}) \leq ab - (2w_{\max}(T \setminus W_{1-\delta}) - a)_+(2h_{\max}(T \setminus W_{1-\delta}) - b)_+ \\
\Leftarrow \quad & 4\gamma + 2\varepsilon - 2(1-\delta)h(T \cap W_{1-\delta}) \\
& \qquad = 1 - h(T \cap W_{1-\delta}) - x - (1-2\delta)_+(1 - 2\gamma + h(T \cap W_{1-\delta}) + x)_+ \\
& \qquad = -2(1-\delta)h(T \cap W_{1-\delta}) - 2x + 2\delta x + 2\gamma + 2\delta - 4\gamma\delta \\
\Leftarrow \quad & 4\gamma + 2\varepsilon = -2x + 2\delta x + 2\gamma + 2\delta - 4\gamma\delta \\
\Leftarrow \quad & \gamma + 2\gamma\delta = -x + \delta x + \delta - \varepsilon \\
\Leftarrow \quad & \gamma = \frac{\delta(1+x) - x - \varepsilon}{1 + 2\delta}.
\end{aligned}$$

So far we assumed the knowledge of $\delta \in ((2x+\varepsilon)/(1-x), 1/2]$ for which we have $h(W_{1-\delta}) \leq f(\delta)$. It is easy to see that this value can be computed by calculating $h(W_{1-\delta})$ for $\delta = 1 - w_i$ for all $r_i = (w_i, h_i)$ with $w_i > 1/2$. As $h(W_{1-\delta})$ changes only for these values of δ, we will necessarily find a suitable δ if one exists. We therefore showed the following lemma.

Lemma 5. *For any fixed $\varepsilon > 0$, there exists a polynomial-time algorithm that, given an instance I with $\mathrm{OPT}(I) = 1$ and $h(W_{1-\delta}) \leq f(\delta)$ for some $\delta \in ((2x+\varepsilon)/(1-x), 1/2]$, returns a packing of I into two bins such that only a height of $1 - x$ is used in the second bin.*

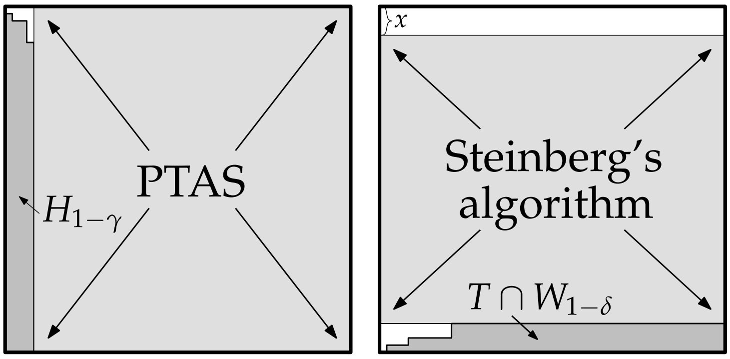


Figure 2: Packing with small total height of $(1-\delta)$-wide items

3.1.2 USING AN AREA GUARANTEE FOR THE WIDE ITEMS

In this section we describe how to use a guarantee on the total area of the wide items for the instances that cannot be packed into a strip of height $2-x$ by Lemma 5. Assume that we have an additional non-trivial area guarantee of ξ for the wide items. Namely,

$$\mathcal{A}(W) \geq \xi + \frac{h(W)}{2} \qquad \text{for } 0 < \xi \leq 1/6. \tag{3.1}$$

We consider two cases in which we use the lower bound from Inequality (3.1) to derive a packing into a strip of height $2-2\xi$.

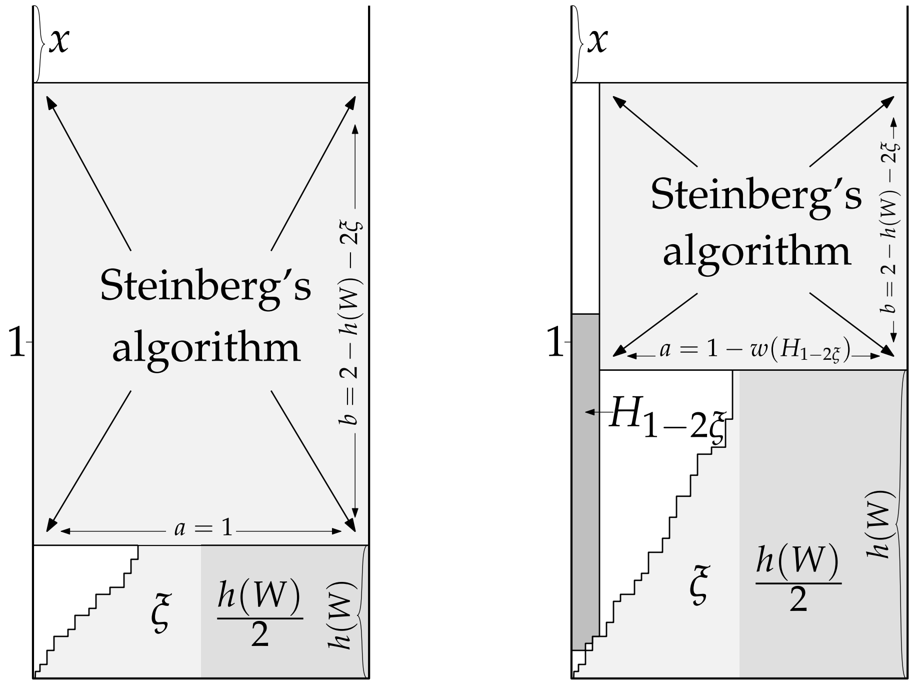
CASE 1. $2-h(W)-2\xi \geq 1$.
Stack the wide items in the bottom of the strip and use Steinberg's algorithm to pack $I \setminus W$ above this stack into a rectangle of size (a,b) with $a=1$ and $b=2-h(W)-2\xi$ (see Figure 3a). Steinberg's algorithm is applicable since we have $h_{\max}(I \setminus W) \leq 1 \leq b$, $w_{\max}(I \setminus W) \leq 1/2 = a/2$ and

$$\begin{aligned} 2\mathcal{A}(I \setminus W) &\leq 2-2\xi-h(W) \\ &= ab = ab - (2w_{\max}(I \setminus W)-a)_+ (2h_{\max}(I \setminus W)-b)_+. \end{aligned}$$

The total height of the packing is $h(W)+2-h(W)-2\xi = 2-2\xi$.

CASE 2. $2-h(W)-2\xi < 1$.
In this case we cannot apply Steinberg's algorithm to pack $I \setminus W$ into the area of size $(1, 2-h(W)-2\xi)$ above the stack of W as $h_{\max}(I \setminus W)$ might be greater than $2-h(W)-2\xi$.

We have $1-2\xi < h(W) \leq 1$ since 1 is a natural upper bound for the total height of the wide items. Pack the items of W in a stack aligned with the bottom right corner of the strip as before. Pack the items of $H_{1-2\xi} \setminus W$ in a stack aligned with the left side of the strip and move this strip downwards as far as possible (see Figure 3b). Corollary 3 shows that $H_{1-2\xi} \setminus W$ can be moved down such that the total height of the packing so



(a) Case 1

(b) Case 2

Figure 3: Using the area guarantee

far is at most $1 + h(W)/2 \leq 3/2$. Let $T = I \setminus (W \cup H_{1-2\xi})$ be the set of the remaining items. We have

$$\mathcal{A}(T) \leq 1 - \mathcal{A}(W) - \mathcal{A}(H_{1-2\xi} \setminus W) \leq 1 - \xi - \frac{h(W)}{2} - (1-2\xi)w(H_{1-2\xi} \setminus W).$$

Pack T with Steinberg's algorithm in the rectangle of size (a, b) with $a = 1 - w(H_{1-2\xi} \setminus W)$ and $b = 2 - h(W) - 2\xi$ above W and to the right of $H_{1-2\xi} \setminus W$. We have $w(H_{1-2\xi} \setminus W) \leq 1/2$ as otherwise all wide items are either above or below an item from $H_{1-2\xi} \setminus W$ in any optimal packing and thus $h(W) \leq 4\xi$ (which is a contradiction to $2 - h(W) - 2\xi < 1$ for $\xi \leq 1/6$). Thus we have $h_{\max}(T) \leq 1 - 2\xi \leq b$ and $w_{\max}(T) \leq 1/2 \leq a$ and with

$$\begin{aligned} &ab - (2w_{\max}(T) - a)_+(2h_{\max}(T) - b)_+ \\ &\geq 2 - h(W) - 2\xi - 2w(H_{1-2\xi} \setminus W) + h(W)w(H_{1-2\xi} \setminus W) + 2\xi w(H_{1-2\xi} \setminus W) \\ &\qquad - (w(H_{1-2\xi} \setminus W))_+(h(W) - 2\xi)_+ \\ &= 2 - h(W) - 2\xi - 2w(H_{1-2\xi} \setminus W) + 4\xi w(H_{1-2\xi}) \end{aligned}$$

we get $2\mathcal{A}(T) \leq ab - (2w_{\max} - a)_+(2h_{\max} - b)_+$. We have $(h(W) - 2\xi)_+ = h(W) - 2\xi$ in the last step of the calculation since $h(W) > 4\xi$. The total height of the packing is dominated by the height of the wide items $h(W)$ plus the height of the target area for Steinberg's algorithm $b = 2 - h(W) - 2\xi$. In total we have a height of $h(W) + 2 - h(W) - 2\xi = 2 - 2\xi$.

In total we showed the following lemma.

Lemma 6. *Let* $0 < \xi \leq 1/6$ *be a constant. If* $\mathcal{A}(W) > \xi + h(W)/2$*, then we can derive a packing into a strip of height* $2 - 2\xi$ *in time* $\mathcal{O}(n \log^2 n)/\log\log n)$.

In Section 3.2 we use this lemma with a value of ξ close to $1/6$ to get a packing into a strip of height close to $5/3$. Here, we show how to derive a bound for ξ for those instances that cannot be packed into a strip of height $2 - x$ by Lemma 5.

Consider a strip with the lower left corner at the origin of a cartesian coordinate system and consider the stack of wide items ordered by non-increasing width and aligned with the lower right corner of the strip. If there exists a $\delta \in ((2x + \varepsilon)/(1 - x), 1/2]$ such that $h(W_{1-\delta}) \leq f(\delta)$ then we use the algorithm of Lemma 5 to pack the instance into a strip of height $2 - x$ (see Figure 2). Otherwise the stack of wide items exceeds the function $f(\delta)$ for *all* $\delta \in ((2x + \varepsilon)/(1 - x), 1/2]$ (see Figure 4). Then we have

$$\mathcal{A}(W) > \int_{\frac{2x+\varepsilon}{1-x}}^{1/2} \frac{\delta(1-x) - 2x - \varepsilon}{1 + 2\delta} d\delta + \frac{h(W)}{2}.$$

In the following we show that for $\xi(x) := \frac{1}{4}(1 - \ln 2) - \frac{1}{4}x(1 + 3\ln 2) - \frac{1}{2}\varepsilon \ln 2$ we have

$$\mathcal{A}(W) > \xi(x) + \frac{h(W)}{2}. \tag{3.2}$$

Using

$$\ln(1 + x) \geq x - \frac{x^2}{2} \tag{3.3}$$

we have

$$\begin{aligned}
&\int_{\frac{2x+\varepsilon}{1-x}}^{1/2} \frac{\delta(1-x) - 2x - \varepsilon}{1 + 2\delta} d\delta \\
&= \left[\frac{1}{4}(2\delta + 1)(1 - x) - \frac{1}{4}(1 + 3x + 2\varepsilon)\ln(2\delta + 1)\right]_{\delta = \frac{2x+\varepsilon}{1-x}}^{\delta = 1/2} \\
&= \frac{1}{2} - \frac{x}{2} - \frac{1}{4}(1 + 3x + 2\varepsilon)\ln 2 - \frac{1}{4}\left(\frac{4x + 2\varepsilon}{1 - x} + 1\right)(1 - x) \\
&\qquad + \frac{1}{4}(1 + 3x + 2\varepsilon)\ln\left(\frac{4x + 2\varepsilon}{1 - x} + 1\right) \\
&\geq \frac{1}{2} - \frac{x}{2} - \frac{\ln 2}{4} - x\frac{3\ln 2}{4} - \varepsilon\frac{\ln 2}{2} - \frac{4x + 2\varepsilon}{4} - \frac{1}{4} + \frac{x}{4} \\
&\qquad + \frac{1}{4}(1 + 3x + 2\varepsilon)\left(4x + 2\varepsilon - \frac{(4x + 2\varepsilon)^2}{2}\right) \\
&\qquad \text{as } \ln\left(\frac{4x + 2\varepsilon}{1 - x} + 1\right) \geq \ln(4x + 2\varepsilon + 1) \text{ and by (3.3)} \\
&> \frac{1}{2} - \frac{x}{2} - \frac{\ln 2}{4} - x\frac{3\ln 2}{4} - \varepsilon\frac{\ln 2}{2} - \frac{4x + 2\varepsilon}{4} - \frac{1}{4} + \frac{x}{4} + \frac{4x + 2\varepsilon}{4} \qquad (3.4) \\
&> \frac{1 - \ln 2}{4} - x\left(\frac{1 + 3\ln 2}{4}\right) - \varepsilon\frac{\ln 2}{2} \\
&= \xi(x)
\end{aligned}$$

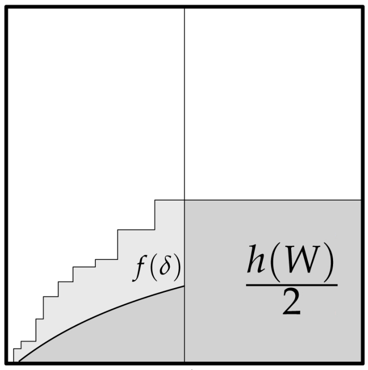


Figure 4: If Lemma 5 is not applicable the stack of wide items exceeds $f(\delta)$, giving an additional area guarantee of $\xi(x)$

The step to (3.4) follows with

$$\begin{aligned}(1+3x+2\varepsilon)\left(4x+2\varepsilon-\frac{(4x+2\varepsilon)^2}{2}\right) &= (1+3x+2\varepsilon)(4x+2\varepsilon-8x^2-8\varepsilon x-2\varepsilon^2)\\ &= 4x+2\varepsilon+2\varepsilon^2-4\varepsilon^3+\varepsilon x(6-22\varepsilon)\\ &\qquad +x^2(4-40\varepsilon-24x)\\ &> 4x+2\varepsilon\end{aligned}$$

as $2\varepsilon^2-4\varepsilon^3 > 0$, $6-22\varepsilon > 0$ and $4-40\varepsilon-24x > 0$ since $\varepsilon < 10^{-5}/2$ and $x < 1/6-5/3\,\varepsilon$.

To finally calculate the approximation ratio that we achieve observe that with the methods of Lemma 5 we derive a packing into a strip of height $2-x$ and with the methods of Lemma 6 we derive a packing into a strip of height $2-2\xi(x)$. Thus we require $2\xi(x) \geq x$ to achieve a packing height of $2-x$. This is satisfied for

$$x \leq \frac{1-\ln 2}{3+3\ln 2} - \varepsilon \leq \frac{1-\ln 2-2\varepsilon\ln 2}{3+3\ln 2}.$$

We can choose $x = (1-\ln 2)/(3+3\ln 2)-\varepsilon$ and together with Lemmas 3, 5 and 6 we proved Theorem 3.

A striking feature of our algorithm is that in many cases it derives a packing where we can make a horizontal cut at height 1 without intersecting any item (or rather at about half of the packings height after scaling the packing back). This feature is very useful for deriving a bin packing algorithm—as we will see in Section 5. On the other hand, it limits the possible packings and hinders us to achieve a better approximation ratio. In the following section we will show that a significantly better approximation ratio can be achieved by lifting this limitation. As the next algorithm also requires much more effort to achieve, we still consider the 1.9396-approximation algorithm an interesting result, especially as it was the first approximation ratio to break the bound of 2.

3.2 AN ALMOST 5/3-APPROXIMATION ALGORITHM FOR STRIP PACKING

In this section we describe our second strip packing algorithm that achieves an approximation ratio arbitrarily close to 5/3. This is the currently best approximation algorithm for strip packing from the perspective of the absolute approximation ratio.

3.2.1 OVERVIEW

Let $\varepsilon < 1/(28 \cdot 151)$ throughout this section. By Lemma 3 we know that we can concentrate on approximating instances that fit into a strip of height 1 and therefore assume $\mathrm{OPT}(I) \leq 1$ for the remainder of this section. The overall approach for our algorithm for strip packing is as follows.

First, we use the methods of Section 3.1.2 to solve instances I with $h(W_{1-130\varepsilon}) \geq 1/3$ and some other direct method to solve instances I with $w(H_{2/3}) \geq 27/28$. Having sufficiently many high or wide items makes it much easier to pack all items without wasting much space. We can easily check these conditions in time $\mathcal{O}(n)$.

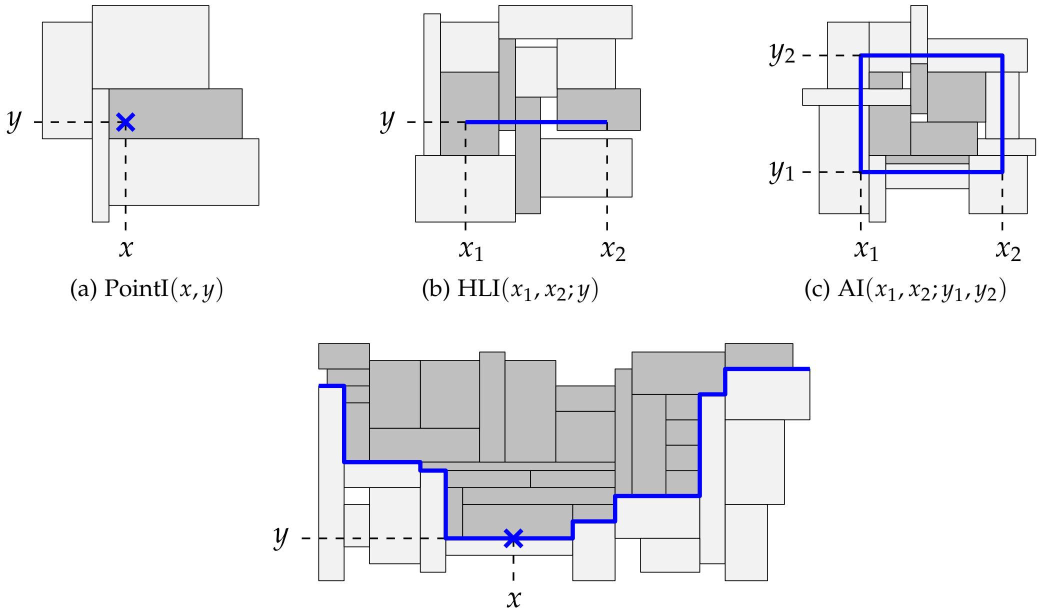
For any instance I that does not satisfy these conditions, we first apply the $\mathcal{PTAS}$ from [5] with an accuracy of $\delta = \varepsilon^2/2$ to pack most of the items into a strip of height 1. Denote the resulting packing of $I' \subseteq I$ by P and let $R = I \setminus I'$ be the set of remaining items. By Theorem 2 we have $\mathcal{A}(R) \leq \varepsilon^2/2 \cdot \mathrm{OPT}_{2-\mathrm{KP}}(I) = \varepsilon^2/2 \cdot \mathcal{A}(I) \leq \varepsilon^2/2$. We are faced with the challenge to combine these remaining rectangles with the packing generated by the $\mathcal{PTAS}$. Pack $R \cap H_{\varepsilon/2}$ into a container $C_1 = (\varepsilon, 1)$ (by forming a stack of the items of total width at most $\mathcal{A}(R)/(\varepsilon/2) \leq \varepsilon$) and pack $R \setminus H_{\varepsilon/2}$ with Steinberg's algorithm into a container $C_2 = (1, \varepsilon)$ (this is possible by Theorem 1 since $h_{\max}(R \setminus H_{\varepsilon/2}) \leq \varepsilon/2$, $w_{\max}(R \setminus H_{\varepsilon/2}) \leq 1$ and $2\mathcal{A}(R \setminus H_{\varepsilon/2}) \leq \varepsilon^2 < \varepsilon$).

Finally, we modify the packing P to free a gap of width ε and height 1 to insert the container C_1 while retaining a total packing height of at most 5/3. This is the main part of our work. Afterwards, we pack C_2 above the entire packing, achieving a total height of at most $5/3 + \varepsilon$.

The running time of the $\mathcal{PTAS}$ from [5] is polynomial in the number of items in the input (it is not explicitly stated in [5]). In the following sections that build upon packing P we only give the additional running time for modifying the packing.

3.2.2 MODIFYING PACKINGS

Our methods involve modifying existing packings in order to insert some additional items. To describe these modifications or, more specifically, the items involved in these modifications, we introduce the following notations—see Figure 5. Let $\mathrm{PointI}(x, y)$ be the item that contains the point (x, y) (in its interior). We use the notation of *vertical line items* $\mathrm{VLI}(x; y_1, y_2)$ and *horizontal line items* $\mathrm{HLI}(x_1, x_2; y)$ as the items that contain any point of the given vertical or horizontal line in their interiors, respectively. Finally, we introduce two notations for items whose interiors are completely contained in a designated area, namely $\mathrm{AI}(x_1, x_2; y_1, y_2)$ for items completely inside the respective



(a) $\text{PointI}(x, y)$ (b) $\text{HLI}(x_1, x_2; y)$ (c) $\text{AI}(x_1, x_2; y_1, y_2)$

(d) $\text{AI}(\text{VPCE}(x, y))$

Figure 5: Notations

rectangle and $\text{AI}(p)$ for items completely above a given polygonal line p, where p is a staircase-function on $[0, 1]$.

To describe such a polygonal line p we define the *vertical polygonal chain extension* of a point (x, y) inside a given packing P as follows. Start at position (x, y) and move leftwards until hitting an item r_i. Then move upwards to the top of r_i, that is, up to position y_i'. Repeat the previous steps until hitting the left side of the strip. Then do the same thing to the right starting again at (x, y). We denote the polygonal chain that results from this process by $\text{VPCE}(x, y)$. In addition, let $\text{VPCE}_{\text{left}}(x, y)$ and $\text{VPCE}_{\text{right}}(x, y)$ be the left and right parts of this polygonal chain, respectively. Another way to describe a polygonal line is by giving a sequence of points, which we denote as $\text{PL}((x_1, y_1), (x_2, y_2), \ldots)$.

We now present the different methods that we apply on the packing P in Sections 3.2.3–3.2.9. In Section 3.2.10 we finally bring the parts together and give the overall algorithm.

3.2.3 TOTAL AREA OF VERY WIDE ITEMS IS LARGE

In this case we apply the methods of our 1.9396-approximation algorithm from Section 3.1.2 as follows.

Lemma 7. *If* $h(W_{1-130\varepsilon}) \geq 1/3$, *then we can derive a packing into a strip of height* $5/3 + 260\varepsilon/3$ *in time* $\mathcal{O}((n \log^2 n)/\log\log n)$.

Proof. Let $W' = W_{1-130\varepsilon}$. The total height of the items in W' gives us a non-trivial additional area guarantee for the wide items as follows.

$$\begin{aligned}A(W) &= A(W') + A(W \setminus W')\\ &> h(W') \cdot (1 - 130\varepsilon) + h(W \setminus W') \cdot 1/2\\ &= h(W') \cdot (1/2 - 130\varepsilon) + h(W)/2\\ &\geq 1/3 \cdot (1/2 - 130\varepsilon) + h(W)/2.\end{aligned}$$

For $\xi = 1/3 \cdot (1/2 - 130\varepsilon)$ we have $A(W) > \xi + h(W)/2$ and $0 < \xi \leq 1/6$. Thus we can apply the methods of Lemma 6 to derive a packing into a strip of height $2 - 2\xi = 5/3 + 260\varepsilon/3$. □

3.2.4 LARGE TOTAL WIDTH OF THE 2/3-HIGH ITEMS

In this section we assume that $w(H_{2/3}) \geq 27/28$, i.e., the total width of the 2/3-high items is very large. As in the previous section we can solve this case directly without using the $\mathcal{PTAS}$ from [5].

For the ease of presentation, let $\alpha = w(H_{2/3}) \geq 27/28$. Since the total height of the items of $W_{1-\alpha/2} \setminus H_{2/3}$ plays an important role in our method, we introduce the notation $y = h(W_{1-\alpha/2} \setminus H_{2/3})$. Moreover, we use the stronger area guarantee of the 5/6-high items and therefore denote their total width by $\beta = w(H_{5/6})$. Finally, let $\delta = w(H_{1/3} \setminus H)$ be the total width of the items of height within 1/3 and 1/2.

BOUNDING y AND δ. Let $\alpha' < \alpha$ such that $W_{1-\alpha/2} = \{r_i \mid w_i > 1 - \alpha/2\} = \{r_i \mid w_i \geq 1 - \alpha'/2\}$, e.g., set α' such that the shortest item in $W_{1-\alpha/2}$ has width $1 - \alpha'/2$. Note that in any optimal packing, all items from $W_{1-\alpha/2}$ occupy the x-interval $(\alpha'/2, 1 - \alpha'/2)$ of width $1 - \alpha'$ completely. On the other hand, there has to be an item from $H_{2/3}$ that intersects this interval since $w(H_{2/3}) = \alpha > \alpha'$. Therefore we have

$$y = h(W_{1-\alpha/2} \setminus H_{2/3}) < 1/3. \qquad (3.5)$$

It follows directly that the sets $W_{1-\alpha/2} \setminus H_{2/3}$ and $H_{1/3}$ are disjoint.

Similarly, only in a total width of $1 - \alpha$ can items in $H_{1/3} \setminus H_{2/3}$ possibly be packed. Since at most two such items can fit on top of each other, a total width of at most $2(1 - \alpha)$ of such items can exist, which is less than $\alpha/2$ by direct calculation for $\alpha > 4/5$. Hence we get

$$\delta \leq 2(1 - \alpha) < \alpha/2. \qquad (3.6)$$

In the following we distinguish three main cases according to y and β. See Figure 6a for the first two cases and Figure 6b for the third case.

CASE 1. $y \geq \frac{4}{3}\frac{1-\alpha}{1-\alpha/2}$.
We use the methods of Corollary 3 for $H \cup W_{1-\alpha/2}$, and need a height of at most $1 + y/2$ which is less than 7/6 by Inequality (3.5). Above it, we define a container of width δ and height 1/2 at the left side of the strip where we pack all remaining 1/3-high items, i.e., $H_{1/3} \setminus H$. Next to it we have an area (a, b) of width $a = 1 - \delta$

and height $b = 2/3 - y/2 > 1/2$. In it we pack all remaining items, noted by $T = I \setminus (H_{1/3} \cup W_{1-\alpha/2})$, that have height at most $h_{\max}(T) \leq 1/3 < b$, width at most $w_{\max}(T) \leq 1 - \alpha/2 < 1 - \delta = a$ by Inequality (3.6), and area at most

$$\mathcal{A}(T) \leq 1 - \mathcal{A}(H_{2/3}) - \mathcal{A}(W_{1-\alpha/2} \setminus H_{2/3}) - \mathcal{A}(H_{1/3} \setminus H) \leq 1 - \frac{2}{3}\alpha - \left(1 - \frac{\alpha}{2}\right) y - \frac{\delta}{3}.$$

This works according to the Steinberg condition for any $y \geq \frac{4}{3}\frac{1-\alpha}{1-\alpha/2}$ since

$$\begin{aligned} ab - (2w_{\max}(T) - a)_+ (2h_{\max}(T) - b)_+ &= (1-\delta)\left(\frac{2}{3} - \frac{y}{2}\right) - (1 - \alpha + \delta)_+ \left(\frac{y}{2}\right)_+ \\ &= \frac{2}{3} - \frac{2\delta}{3} - y + \frac{\alpha y}{2} \\ &= \frac{2}{3} - \frac{2\delta}{3} - 2y + \alpha y + y(1 - \frac{\alpha}{2}) \\ &\geq \frac{2}{3} - \frac{2\delta}{3} - 2y + \alpha y + \frac{4}{3}\frac{1-\alpha}{1-\alpha/2}(1 - \frac{\alpha}{2}) \\ &= 2\left(1 - \frac{2}{3}\alpha - (1 - \frac{\alpha}{2})y - \frac{\delta}{3}\right) \\ &\geq 2\mathcal{A}(T). \end{aligned}$$

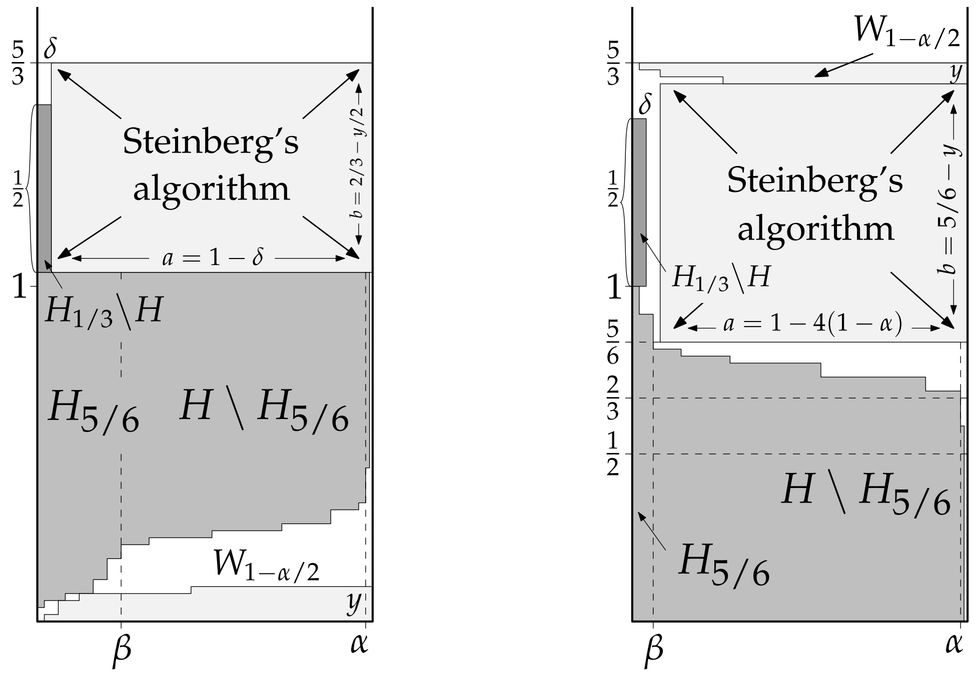
CASE 2. $\beta \geq 4(1-\alpha)$.
We use the same packing as in Case 1. The total area of the high items is now at least $\frac{5}{6}\beta + \frac{2}{3}(\alpha - \beta) = \frac{2}{3}\alpha + \frac{1}{6}\beta \geq 2/3$. Therefore, the remaining unpacked items, noted by T, have area at most

$$\mathcal{A}(T) \leq 1 - \mathcal{A}(H_{2/3}) - \mathcal{A}(W_{1-\alpha/2} \setminus H_{2/3}) - \mathcal{A}(H_{1/3} \setminus H) \leq \frac{1}{3} - \left(1 - \frac{\alpha}{2}\right) y - \frac{\delta}{3}.$$

Since only the area of T changes compared to Case 1, we only have to verify the third Steinberg condition to pack T with Steinberg's algorithm into the area (a, b).

$$\begin{aligned} ab - (2w_{\max}(T) - a)_+ (2h_{\max}(T) - b)_+ &= \frac{2}{3} - \frac{2\delta}{3} - y + \frac{\alpha y}{2} \\ &= \frac{2}{3} - \frac{2\delta}{3} - (2-\alpha)y + (1 - \frac{\alpha}{2})y \\ &\geq 2\left(\frac{1}{3} - \frac{\delta}{3} - (1 - \frac{\alpha}{2})y\right) \\ &\geq 2\mathcal{A}(T). \end{aligned}$$

CASE 3. $y < \frac{4}{3}\frac{1-\alpha}{1-\alpha/2}$ and $\beta < 4(1-\alpha)$.
Note that $y < \frac{4}{3}\frac{1-\alpha}{1-\alpha/2} \leq \frac{4}{3}\frac{1-27/28}{1-27/56} = \frac{56}{609} < \frac{1}{6}$ in this case. We pack the set H of the high items aligned with the bottom of the strip, sorted by non-increasing heights (from left to right). We pack the items of $W_{1-\alpha/2} \setminus H$ stacked in the area $[0, 1] \times [5/3 - y, 5/3]$ and the items of $H_{1/3} \setminus H$ in the area $[0, \delta] \times [1, 3/2]$ (this is possible because $3/2 \leq 5/3 - y$). We have $\delta < 4(1-\alpha)$, since by Inequality (3.6) we have $\delta < 2(1-\alpha)$. Furthermore, by assumption we have $\beta < 4(1-\alpha)$. It follows that the area $[4(1-\alpha), 1] \times [5/6, 5/3 - y]$ of width $a = 1 - 4(1-\alpha) = 4\alpha - 3$ and height $b = 5/3 - y - 5/6 \geq 2/3$ is still free.



(a) Case 1 or 2. $y \geq \frac{4}{3}\frac{1-\alpha}{1-\alpha/2}$ or $\beta \geq 4(1-\alpha)$

(b) Case 3. $y < \frac{4}{3}\frac{1-\alpha}{1-\alpha/2}$ and $\beta < 4(1-\alpha)$

Figure 6: Packing methods for Lemma 8

We pack all remaining items, noted by T, in this area using Steinberg's algorithm. We have $h_{\max}(T) \leq 1/3 \leq b/2$, $w_{\max}(T) \leq 1 - \alpha/2 < 4\alpha - 3 = a$, since $\alpha > 8/9$, and area at most

$$\mathcal{A}(T) \leq 1 - \mathcal{A}(H_{2/3}) - \mathcal{A}(W_{1-\alpha/2} \setminus H_{2/3}) \leq 1 - \frac{2}{3}\alpha - \left(1 - \frac{\alpha}{2}\right)y.$$

Hence the Steinberg condition is satisfied for $\alpha \geq 27/28 \geq (27 - 30y)/(28 - 30y)$ since

$$\begin{aligned}
&ab - (2w_{\max}(T) - a)_+(2h_{\max}(T) - b)_+ \\
&\quad = ab \\
&\quad = (4\alpha - 3)\left(\frac{5}{6} - y\right) \\
&\quad = -\frac{4\alpha}{3} + y\alpha + \alpha\left(\frac{28 - 30y}{6}\right) - \frac{5}{2} + 3y \\
&\quad \geq -\frac{4\alpha}{3} + y\alpha + \frac{27 - 30y}{28 - 30y} \cdot \frac{28 - 30y}{6} - \frac{5}{2} + 3y \\
&\quad = -\frac{4\alpha}{3} + y\alpha + 2 - 2y \\
&\quad \geq 2\mathcal{A}(T).
\end{aligned}$$

Since the running time of our methods is dominated by the application of Steinberg's algorithm we showed the following lemma.

Lemma 8. *If $w(H_{2/3}) \geq 27/28$, then we can derive a packing of I into a strip of height $5/3$ in time $\mathcal{O}((n\log^2 n)/\log\log n)$.*

This finishes the presentation of the methods that we directly apply to the input if $h(W_{1-130\varepsilon}) \geq 1/3$ (Section 3.2.3) or $w(H_{2/3}) \geq 27/28$ (Section 3.2.4). In the following sections we always assume that we already derived a packing P using the $\mathcal{PTAS}$ from [5] and it remains to free a place for the containers C_1 and C_2 of size $(\varepsilon, 1)$ and $(1, \varepsilon)$, respectively.

3.2.5 item of height greater than 1/3

Lemma 9. *If the following conditions hold for P, namely*

9.1. there is an item r_1 of height $h_1 > 1/3$ with one side at position $x_1^* \in [\varepsilon, 1/2-\varepsilon]$, and

9.2. the total width of $2/3$-high items to the left of x_1^* is at most $x_1^* - \varepsilon$, that is

$$w(\mathrm{AI}(0, x_1^*; 0, 1) \cap H_{2/3}) \leq x_1^* - \varepsilon,$$

then we can derive a packing of I into a strip of height $5/3+\varepsilon$ in additional time $\mathcal{O}(n\log n)$.

Note that Condition 9.1 leaves open whether x_1^* refers to the left or right side of r_1 as our method works for both cases. In particular, r_1 could be one of the $2/3$-high items from Condition 9.2.

Proof. Assume w.l.o.g. $y_1' > 2/3$ by otherwise mirroring the packing P over $y = 1/2$.

We lift up a part of the packing P in order to derive a gap of sufficient height to insert the container C_1. In this case we mirror the part of the packing that we lift up. Algorithm 1 gives a compressed version of the following detailed description. See Figure 7 for an illustration.

Consider the contour C_{lift} defined by a horizontal line at height $y = y_1' - 1/3$ to the left of x_1^*, a vertical line at width $x = x_1^*$ up to y_1' and a vertical polygonal chain extension to the right starting at the top of r_1. More formally,

$$C_{\mathrm{lift}} = \mathrm{PL}\Big((0, y_1' - \frac{1}{3}), (x_1^*, y_1' - \frac{1}{3}), (x_1^*, y_1')\Big) + \mathrm{VPCE}_{\mathrm{right}}(x_1^*, y_1'),$$

where the $+$-operator denotes the concatenation of polygonal lines (see thick line in Figure 7a). Let $T = \mathrm{AI}(C_{\mathrm{lift}})$ be the set of items that are completely above this contour.

Move up T by $2/3$ (and hereby move T completely above the previous packing since $y_1' > 2/3$ and thus $y_1' - 1/3 > 1/3$) and mirror T vertically, i.e., over $x = 1/2$. Let y_{bottom} be the height of C_{lift} at $x = 1/2$ (C_{lift} crosses the point $(1/2, y_{\mathrm{bottom}})$). By definition, C_{lift} is non-decreasing and no item intersects with C_{lift} to the right of x_1^*. Therefore, T is completely packed above $y = y_{\mathrm{bottom}} + 2/3$ on the left side of the strip, i.e., for $x \leq 1/2$, and $P \setminus T$ does not exceed y_{bottom} between $x = x_1^*$ and $x = 1/2$. Thus between $x = x_1^*$ and $x = 1/2$ we have a gap of height at least $2/3$.

Let $B = \mathrm{HLI}(0, x_1^*; y_1' - 1/3)$ be the set of items that intersect height $y = y_1' - 1/3$ to the left of x_1^* (see Figure 7a). Note that $r_1 \in B$, if x_1^* corresponds to the right side of r_1. Remove B from the packing, order the items by non-increasing order of height and build a top-left-aligned stack at height $y = y_{\text{bottom}} + 2/3$ and distance ε from the left side of the strip. Since we keep a slot of width ε to the left, the stack of B might exceed beyond x_1^*. This overhang does not cause an overlap of items because Condition 9.1 ensures that $x_1^* \leq 1/2 - \varepsilon$ and thus the packing of B does not exceed position $x = 1/2$ and Condition 9.2 ensures that the excessing items have height at most $2/3$ whereas the gap has height at least $2/3$.

Now pack the container C_1 top-aligned at height $y_{\text{bottom}} + 2/3$ directly at the left side of the strip. C_1 fits here since $y_{\text{bottom}} + 2/3 - (y_1' - 1/3) = 1 + y_{\text{bottom}} - y_1' \geq 1$. Finally, pack C_2 above the entire packing at height $y = 5/3$, resulting in a total packing height of $5/3 + \varepsilon$. □

Note that Lemma 9 can symmetrically be applied for a $1/3$-high item with one side at position $x_1^* \in [1/2 + \varepsilon, 1 - \varepsilon]$ with $w(\text{AI}(x_1^*, 1; 0, 1) \cap H_{2/3}) \leq 1 - x_1^* - \varepsilon$ by mirroring P over $x = 1/2$.

Algorithm 1 Edge of height greater than $1/3$

Requirement: Packing P that satisfies Conditions 9.1 and 9.2 and $y_1' > 2/3$.

1: Move up the items $T = \text{AI}(C_{\text{lift}})$ by $2/3$ and then mirror the packing of these items vertically at position $x = 1/2$.

2: Order the items of $B = \text{HLI}(0, x_1^*; y_1' - 1/3)$ by non-increasing order of height and pack them into a top-aligned stack at position $(\varepsilon, y_{\text{bottom}} + 2/3)$.

3: Pack C_1 top-aligned at position $(0, y_{\text{bottom}} + 2/3)$ and pack C_2 above the entire packing.

3.2.6 NO 1/3-HIGH ITEMS CLOSE TO THE SIDE OF THE BIN

Lemma 10. *Let $c_1 > 0$ be a constant. If the following conditions hold for P, namely*

10.1. there is no $1/3$-high item that intersects $[c_1\varepsilon, (c_1 + 1)\varepsilon] \times [0, 1]$, and

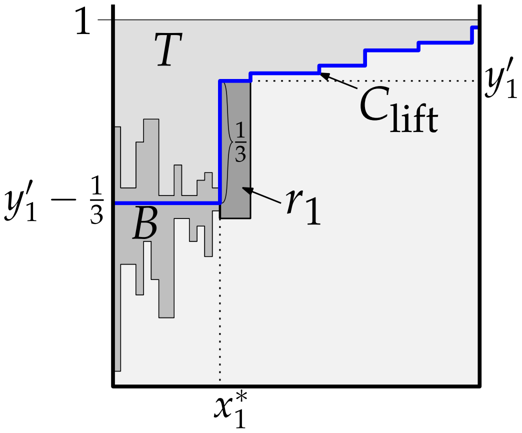
10.2. we have $h(W_{1-5(c_1+1)\varepsilon}) < 1/3$,

then we can derive a packing of I into a strip of height $5/3 + \varepsilon$ in additional time $\mathcal{O}(n)$.

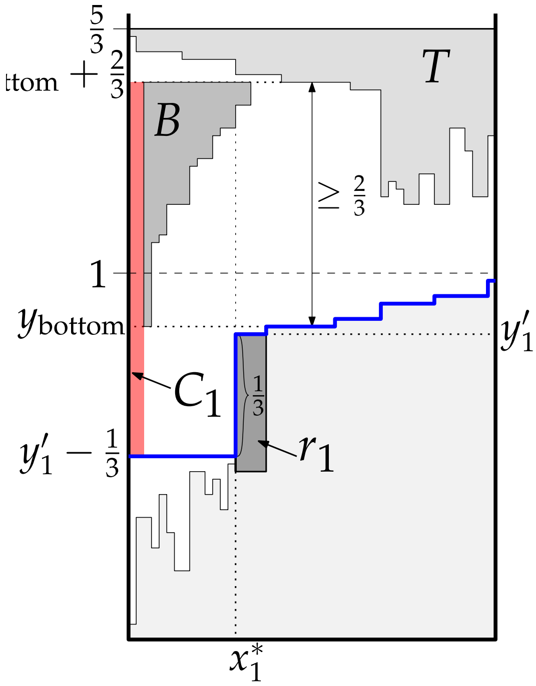
Proof. Let $W' = W_{1-5(c_1+1)\varepsilon} \cap \text{VLI}((c_1 + 1)\varepsilon; 0, 1)$ be the set of rectangles of width larger than $1 - 5(c_1 + 1)\varepsilon$ intersecting the vertical line $x = (c_1 + 1)\varepsilon$. By Condition 10.2 we have $h(W') < 1/3$.

Consider the rectangle $r_1 = \text{PointI}((c_1 + 1)\varepsilon, 1/2)$ (if no such rectangle exists, we set r_1 as a dummy rectangle of height and width equal to 0). We have to distinguish two cases depending on this rectangle and the set W', or to be more accurate their amount of heights above and below the horizontal line at height $y = 1/2$. Therefore, let $a = 1/2 - y_1$ be the height of r_1 below $y = 1/2$ and $a' = y_1' - 1/2$ the height above $y = 1/2$. Furthermore, let $h = h(W' \cap \text{VLI}((c_1 + 1)\varepsilon; 0, y_1))$ and $h' = h(W' \cap \text{VLI}((c_1 + 1)\varepsilon; y_1', 1))$ be the heights of W' above and below $y = 1/2$ excluding r_1 (if $r_1 \in W'$).

Note, that by Condition 10.1 the height of r_1 is $h_1 \leq 1/3$, hence it intersects at most one of the horizontal lines at height $y = 1/3$ or $y = 2/3$.



(a) Definition of C_{lift}, T and B.



(b) Modified packing where C_1 is packed in the red slot of height 1 and width ε.

Figure 7: Packing methods for Lemma 9

We are going to cut a slot of width ε between $c_1\varepsilon$ and $(c_1+1)\varepsilon$ down to a height y_{cut}. The value y_{cut} depends on the particular packing. So we distinguish between two cases:

1. If $a+h \leq 1/6$ or $a'+h' \leq 1/6$, we will assume w.l.o.g that $a'+h' \leq 1/6$ by otherwise mirroring the packing horizontally over $y = 1/2$. In this case we set $y_{\text{cut}} = y'_1$.
2. If $a+h > 1/6$ and $a'+h' > 1/6$, we will assume w.l.o.g that $y_1 \geq 1/3$ by otherwise mirroring the packing horizontally over $y = 1/2$. Here we set $y_{\text{cut}} = y_1$.

Note, if $r_1 \in W'$ it follows that we are in the first case, since $h+a+a'+h' = h(W') < 1/3$ and so $h+a < 1/6$ or $h'+a' < 1/6$. In both cases we have $y_{\text{cut}} \in [1/3, 2/3]$.

ALGORITHM. We are going to cut a slot of width ε between $c_1\varepsilon$ and $(c_1+1)\varepsilon$ down to height y_{cut}, which is either y'_1 or y_1 (hence $\text{PointI}((c_1+1)\varepsilon, y_{\text{cut}}) = \emptyset$). Let $X = [c_1\varepsilon, (c_1+1)\varepsilon] \times [y_{\text{cut}}, 1]$ be the designated slot that we want to free. To do this we differentiate four sets of items intersecting X. The entire algorithm is given in Algorithm 2—see Figure 8 for an illustration.

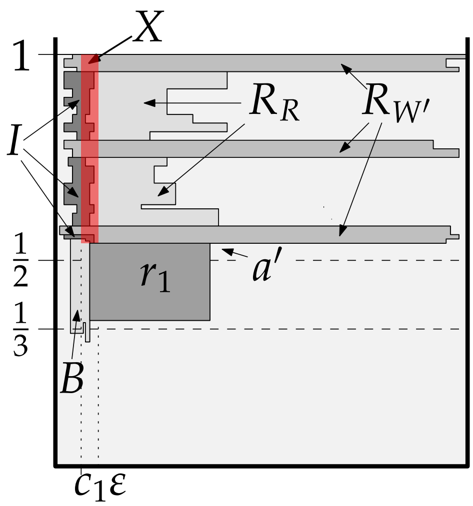
- Let $R_{W'} = \text{VLI}((c_1+1)\varepsilon; y_{\text{cut}}, 1) \cap W'$ be the set of items in W' which intersect X by crossing the vertical line at width $x = (c_1+1)\varepsilon$. Notice, that if $r_1 \in W'$, then $y_{\text{cut}} = y'_1$. Therefore, $R_{W'}$ has total height h'. Place the items of $R_{W'}$ into a container $C_{R_{W'}}$ of height $h' < 1/3$ and width 1 and pack it at position $(0, 5/3 - h')$.

- Let $R_R = \text{VLI}((c_1+1)\varepsilon; y_{\text{cut}}, 1) \setminus W'$ be the set of remaining items intersecting X by crossing the vertical line at width $x = (c_1+1)\varepsilon$. Pack these items left-aligned into a container C_{R_R} of width $1 - 5(c_1+1)\varepsilon$ and height at most $1 - y_{\text{cut}} - h'$. This container is placed at position $(5(c_1+1)\varepsilon, 1)$. This does not cause a conflict, since y_{cut} is always greater than $1/3$ and $h(R_R) + h(R_{W'}) \leq 1 - y_{\text{cut}} \leq 2/3$.

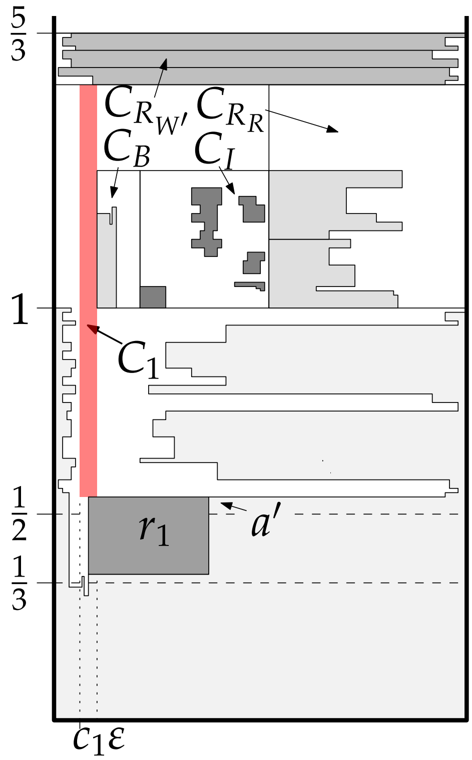
- Let $B = \text{HLI}(c_1\varepsilon, (c_1+1)\varepsilon; y_{\text{cut}})$ be the rectangles which intersect X from the bottom. Note, that there is no rectangle at position $((c_1+1)\varepsilon, y_{\text{cut}})$. By Condition 10.1, these rectangles have height at most $1/3$ and fit bottom-aligned into a container C_B of width $(c_1+1)\varepsilon$ and height $1/3$. Place C_B at position $((c_1+1)\varepsilon, 1)$.

- Let $I = \text{AI}(c_1\varepsilon, (c_1+1)\varepsilon; y_{\text{cut}}, 1) \cup \text{VLI}(c_1\varepsilon; y_{\text{cut}}, 1) \setminus (R_{W'} \cup R_R \cup B)$ be the set of remaining rectangles which are completely inside X or intersect X only from the left. This packing has total height $1 - y_{\text{cut}} \in [1/3, 2/3]$.

  We want to place I between height 1 and $5/3 - h' \geq 4/3$. Therefore, packing I into a container C_I of height $1/3$ is sufficient. To do this we partition I into three sets. Let $I_1 \subseteq I$ be the subset of items that intersect height $y = 2/3$ (these items fit bottom-aligned into a container of size $((c_1+1)\varepsilon, 1/3)$) and let $I_2 \subseteq I$ and $I_3 \subseteq I$ be the subsets of I that lie completely above and below $y = 2/3$, respectively. By preserving the packing of I_2 and I_3 we can pack each into a container $((c_1+1)\varepsilon, 1/3)$. In total we pack I into $C_I = (3(c_1+1)\varepsilon, 1/3)$. This container is placed at position $(2(c_1+1)\varepsilon, 1)$.



(a) Definition of $R_{W'}$, R_R, B and I (here in Case 1 with $a' + h' \leq 1/6$)



(b) Modified packing where C_1 is packed in the red slot of height 1 and width ε.

Figure 8: Packing methods for Lemma 10 (the x-direction is distorted, i.e., ε is chosen very large, to illustrate the different sets that intersect with X)

Finally, we insert C_1 into the free slot X and pack C_2 above the entire packing. We have to prove that the slot has sufficient depth for C_1. The slot starts at height y_{cut} and goes up to $5/3 - h'$. Therefore, we have to check whether $5/3 - h' - y_{\text{cut}} \geq 1$.

In the first case, we have $h' + a' \leq 1/6$ and $y_{\text{cut}} = y_1' = a' + 1/2$. Hence,

$$5/3 - h' - y_{\text{cut}} = 5/3 - h' - a' - 1/2 \geq 5/3 - 1/6 - 1/2 = 1.$$

In the second case, we have $h + a > 1/6$ and $y_{\text{cut}} = y_1 = 1/2 - a$. From our discussion above we know that $h + h' < 1/3$. Hence,

$$\begin{aligned} 5/3 - h' - y_{\text{cut}} &= 5/3 - h' + a - 1/2 \\ &> 5/3 - 1/3 + h + a - 1/2 \\ &\geq 5/3 - 1/3 + 1/6 - 1/2 = 1. \end{aligned}$$ □

Obviously, the methods of Lemma 10 can similarly be applied if there is no 1/3-high item that intersects $[1 - (c_1 + 1)\varepsilon, 1 - c_1\varepsilon] \times [0, 1]$ at the right side of P.

Algorithm 2 No 1/3-high items close to the side of the strip

Requirement: Packing P that satisfies Conditions 10.1 and 10.2.

1: Pack $R_{W'} = \text{VLI}((c_1 + 1)\varepsilon; y_{\text{cut}}, 1) \cap W'$ into a container $C_{R_{W'}} = (1, h')$ at position $(5/3 - h', 0)$.
2: Pack $R_R = \text{VLI}((c_1 + 1)\varepsilon; y_{\text{cut}}, 1) \setminus W'$ into a container $C_{R_R} = (1 - 5(c_1 + 1)\varepsilon, 2/3 - h')$ at position $(5(c_1 + 1)\varepsilon, 1)$.
3: Pack $B = \text{HLI}(c_1\varepsilon, (c_1 + 1)\varepsilon; y_1) \setminus (R_{W'} \cup R_R)$ into a container $C_B = ((c_1 + 1)\varepsilon, 1/3)$ at position $((c_1 + 1)\varepsilon, 1)$.
4: Pack $I = (\text{AI}(c_1\varepsilon, (c_1 + 1)\varepsilon; y_{\text{cut}}, 1) \cup \text{VLI}(c_1\varepsilon; y_{\text{cut}}, 1)) \setminus (R_{W'} \cup R_R \cup B)$ into a container $C_I = (3(c_1 + 1)\varepsilon, 1/3)$ at position $(2(c_1 + 1)\varepsilon, 1)$.
5: Pack C_1 into the slot X at position $(c_1\varepsilon, y_{\text{cut}})$ and pack C_2 above the entire packing.

3.2.7 ONE SPECIAL BIG ITEM IN P

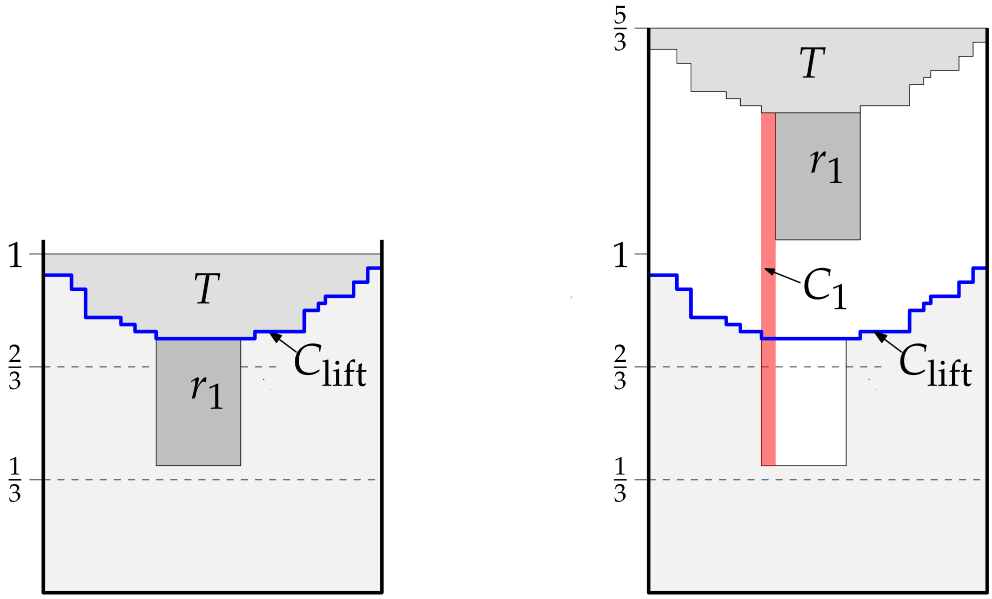
Lemma 11. *If the following condition holds for P, namely*

11.1. there is an item r_1 of height $h_1 \in [1/3, 2/3]$ and width $w_1 \in [\varepsilon, 1 - 2\varepsilon]$, and $y_1 \geq 1/3$ or $y_1' \leq 2/3$,

then we can derive a packing of I into a strip of height $5/3 + \varepsilon$ in additional time $\mathcal{O}(n)$.

Proof. See Figure 9 for an illustration of the following proof. W.l.o.g. we assume that $y_1 \geq 1/3$, by otherwise mirroring the packing horizontally. Furthermore, we assume that $x_1' \leq 1 - \varepsilon$ since $w_1 \leq 1 - 2\varepsilon$ and otherwise mirror the packing vertically, i.e., over $x = 1/2$.

Define a vertical polygonal chain extension $C_{\text{lift}} = \text{VPCE}(x_1, y_1')$ starting on top of r_1 and let $T = \text{AI}(C_{\text{lift}})$. Move up the rectangles in T and the rectangle r_1 by 2/3 and hereby move r_1 completely out of the previous packing, since $y_1 \geq 1/3$. Then move r_1 to the right by ε, this is possible, since $x_1' \leq 1 - \varepsilon$.



(a) Definition of T and C_{lift}.

(b) Modified packing where C_1 is packed in the red slot of height 1 and width ε.

Figure 9: Packing methods for Lemma 11

In the hole vacated by r_1 we have on the left side a free slot of width ε (since $w_1 \geq \varepsilon$ and since we moved r_1 to the right by ε) and height $2/3 + h_1 \geq 1$ (since we moved up T by $2/3$ and since $h_1 \geq 1/3$). Place C_1 in this slot and pack C_2 on top of the packing at height $5/3$. □

Algorithm 3 Rectangle of height $1/3$

Requirement: Packing P that satisfies Condition 11.1.

1: Define $C_{\text{lift}} := \text{VPCE}(x_1, y'_1)$ and move up $T = \text{AI}(C_{\text{lift}})$ by $2/3$.

2: Move up r_1 by $2/3$ and then by ε to the right, i.e., pack r_1 at position $(x_1 + \varepsilon, y_1 + 2/3)$.

3: Pack C_1 into the slot vacated by r_1 and pack C_2 above the entire packing.

Lemma 12. *Let $c_2 > 0$ be a constant. If the following conditions hold for P, namely*

12.1. there is an item r_1 of height $h_1 \in [1/3, 2/3]$ and width $w_1 \in [(4c_2+1)\varepsilon, 1]$ with $y_1 < 1/3$ and with $y'_1 > 2/3$, and

12.2. we have $w(H_{2/3}) \geq 1 - w_1 - c_2\varepsilon$,

then we can derive a packing of I into a strip of height $5/3 + \varepsilon$ in additional time $O(n)$.

Proof. Since the height of r_1 is $h_1 \leq 2/3$ we can assume w.l.o.g. that r_1 does not intersect $y = 1/6$, i.e., $y_1 \geq 1/6$ (by otherwise mirroring over $y = 1/2$).

We want to line up all rectangles in the instance I of height greater than $h = \max(1/2, 1 - h_1)$ and the rectangle r_1 on the bottom of the strip. These rectangles fit there, since in any optimal solution they have to be placed next to each other (all rectangles of $H_h = \{r_i \mid h_i > h\}$ have to intersect the horizontal line at height $y = 1/2$ and no rectangle of H_h fits above r_1). Since $1 - h_1 \geq 1/3$, $H_{2/3}$ is included in the set H_h. See Figure 10 for an illustration of the following algorithm and refer to Algorithm 4 for a compressed description.

Let $T = \text{AI}(0, 1; 2/3, 1)$ be the rectangles which lie completely above the horizontal line at height $y = 2/3$. We move up the rectangles in T by $1/3$ into the area $[0, 1] \times [1, 4/3]$. Now there is a free space of height at least $1/3$ above r_1.

Let $B = \text{AI}(0, 1; 0, 1/3)$ be the rectangles which lie completely below the horizontal line at height $y = 1/3$. We pack these items into a container $C_B = (1, 1/3)$ by preserving the packing of B and pack C_B at position $(0, 4/3)$, i.e., directly above T. Since by assumption r_1 does not intersect the horizontal line at height $y = 1/6$, there is a free space of height at least $1/6$ below r_1.

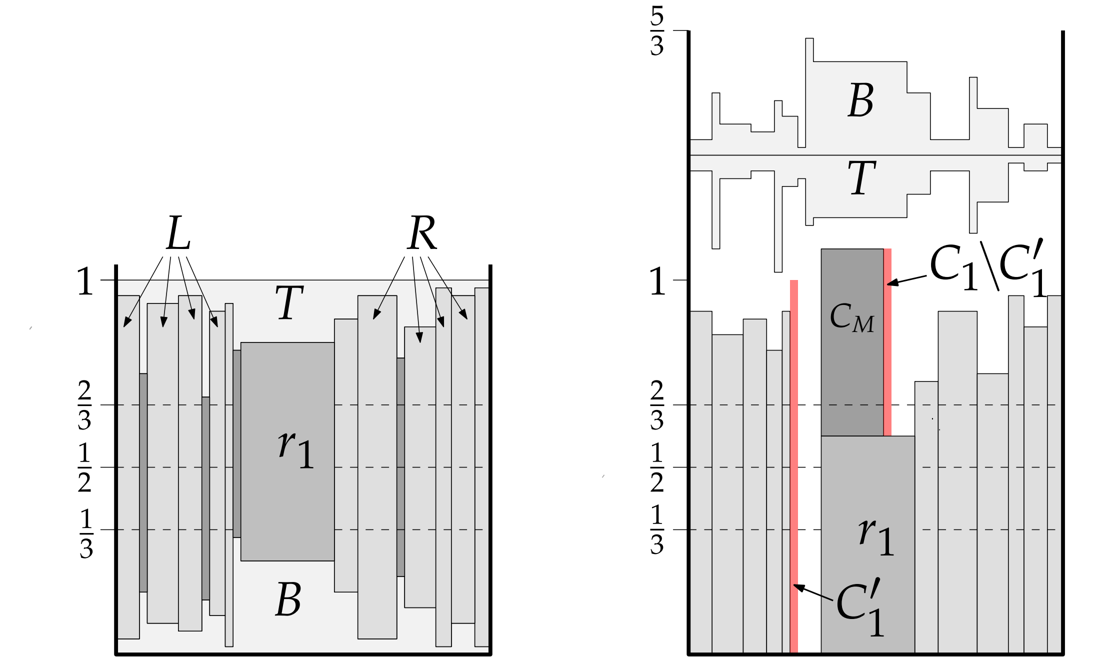
The remaining items of height smaller than h except r_1 have to intersect one of the horizontal lines at height $1/3$ or $2/3$ or lie completely between them. We denote these rectangles by $M_1 = \text{HLI}(0, 1; 1/3) \setminus (H_h \cup \{r_1\})$, $M_2 = \text{HLI}(0, 1; 2/3) \setminus (H_h \cup \{r_1\} \cup M_1)$ and $M_3 = \text{AI}(0, 1; 1/3, 2/3)$. Since each rectangle in $H_{2/3}$ and r_1 intersects both of these lines, the rectangles in $M = M_1 \cup M_2 \cup M_3$ lie between them in slots of total width $c_2\varepsilon$. Therefore, we can pack M_1 and M_2 each bottom-aligned into a container $(c_2\varepsilon, h)$. Furthermore, the rectangles in M_3 fit into a container $(c_2\varepsilon, 1/3)$ by pushing the packing of the slots together. In total we pack M into a container $C_M = (3c_2\varepsilon, h)$ and pack it aside for the moment.

After these steps we removed all rectangles of height at most h except r_1 out of the previous packing. All remaining items intersect the horizontal line at height $y = 1/2$. We line up the rectangles in $L = \text{HLI}(0, x_1; 1/2)$, i.e., the remaining rectangles on the left of r_1, bottom-aligned from left to right starting at position $(0, 0)$. The rectangles in $R = \text{HLI}(0, x_1'; 1/2)$ (the remaining rectangles on the right of r_1) are placed bottom-aligned from right to left starting at position $(1, 0)$. Now move r_1 down to the ground, i.e., pack r_1 at position $(x_1, 0)$. Above r_1 is a free space of height at least $1/2$, since we moved T up by $1/3$ and r_1 down by at least $1/6$. The free space has also height at least $1 - h_1$, since there is no item left above r_1 up to height 1. Hence, in total, this leaves us a free space of width $w_1 \geq (4c_2 + 1)\varepsilon$ and height h. Denote this area by $X = [x, x'] \times [h_1, h_1 + h]$ with $x = x_1$ and $x' = x_1 + w_1$.

Move r_1 to the right by at most $c_2\varepsilon$ until it touches the first rectangle in R, i.e., place r_1 at position $(1 - w(R) - w_1, 0)$. This reduces the width of the free area on top of r_1 to $X' = [x + c_2\varepsilon, x'] \times [h_1, h_1 + h]$. Note, the width of X' is still at least $(3c_2 + 1)\varepsilon$.

In the next step we reorganize the packing of C_1. Recall, that the rectangles in C_1 are placed bottom-aligned in that container. Let C_1' be the rectangles in C_1 of height larger than h. By removing C_1', we can resize the height of C_1 down to h. The resized container C_1 and the container C_M have both height h and total width at most $(3c_2 + 1)\varepsilon$. Place them on top of r_1 in the area X'.

Then place the rectangles in C_1' into the free slot on the left side of r_1. They fit there, since in any optimal packing all rectangles of height greater than h in the instance and



(a) Definition of T, B, L, R and M (dark items squeezed between L and R).

(b) Modified packing where C_1' is packed in the left-hand red slot of height 1 and $C_1 \setminus C_1'$ is packed in the right-hand red slot of height h and width ε.

Figure 10: Packing methods for Lemma 12

r_1 have to be placed next to each other (all rectangles of height greater than h have to intersect the horizontal line at height $y = 1/2$ and none of them fits above r_1). Finally, pack C_2 above the entire packing at height $5/3$. □

3.2.8 TWO RECTANGLES OF HEIGHT BETWEEN 1/3 AND 2/3

Lemma 13. *If the following conditions hold for P, namely*

13.1. there are rectangles r_1, r_2 with heights $h_1, h_2 \in [1/3, 2/3]$ and widths $w_1, w_2 \geq \varepsilon$, and

13.2. we have $y_1 < y_2'$ and $y_2 < y_1'$.

then we can derive a packing of I into a strip of height $5/3 + \varepsilon$ in additional time $\mathcal{O}(n)$.

Proof. See Figure 11 for an illustration of the following algorithm which is given in Algorithm 5. W.l.o.g. let r_1 be the wider rectangle ($w_1 \geq w_2$). Let $C_1^{\text{lift}} = \text{VPCE}(x_1', y_1')$ and $C_2^{\text{lift}} = \text{VPCE}(x_2', y_2')$ be the vertical polygonal chain extensions of the top of r_1 and r_2, respectively. Furthermore, let $T_1 = \text{AI}(C_1^{\text{lift}})$ and $T_2 = \text{AI}(C_2^{\text{lift}})$ be the rectangles above these polygons.

Note that $r_1 \notin T_2$ by Condition 13.2, since otherwise we have $y_1 \geq y_2'$. The same argument holds for the statement $r_2 \notin T_1$.

Algorithm 4 Single big item of height $1/3$

Requirement: Packing P that satisfies Conditions 12.1 and 12.2.

1: Move up $T = \mathrm{AI}(0,1;2/3,1)$ by $1/3$
2: Pack the rectangles in $B = \mathrm{AI}(0,1;0,1/3)$ into a container $C_B = (1,1/3)$ at position $(0,4/3)$.
3: Pack the rectangles in $M = (\mathrm{AI}(0,1;1/3,2/3) \cup \mathrm{HLI}(0,1;1/3) \cup \mathrm{HLI}(0,1;2/3)) \setminus (H_h \cup \{r_1\})$ into a container $C_M = (3c_2, h)$.
4: Line up the rectangles in $L = \mathrm{HLI}(0, x_1; 1/2)$ on the left of r_1 starting at position $(0,0)$.
5: Line up the rectangles in $R = \mathrm{HLI}(0, x_1'; 1/2)$ on the right of r_1 starting at position $(1,0)$ from right to left.
6: Pack r_1 at position $(1 - w(R) - w_1, 0)$, by moving r_1 to the bottom of the strip and at most $c_2\varepsilon$ to the right.
7: Pack C_M and the resized container C_1 on top of r_1 into the area X'.
8: Pack the rectangles $C_1' \subseteq C_1$ of height greater than h into the slot vacated by r_1 and pack C_2 above the entire packing.

Let $T_3 = T_1 \cup T_2$ be the rectangles above r_1 and r_2. We move up the rectangles in T_3 by $2/3$. This leaves a free area of height $2/3$ above r_1 and r_2. We place r_2 directly above r_1 into that free area. This is possible because $w_1 \geq w_2$ and $h_2 \leq 2/3$. The hole vacated by r_2 has width $w_2 \geq \varepsilon$ and height at least 1, since $h_2 \geq 1/3$ and T_3 was moved up by $2/3$. Finally, we place C_1 into that hole and C_2 on top of the packing at height $5/3$. □

Algorithm 5 Two rectangles of height between $1/3$ and $2/3$

Requirement: Packing P that satisfies Conditions 13.1 and 13.2.

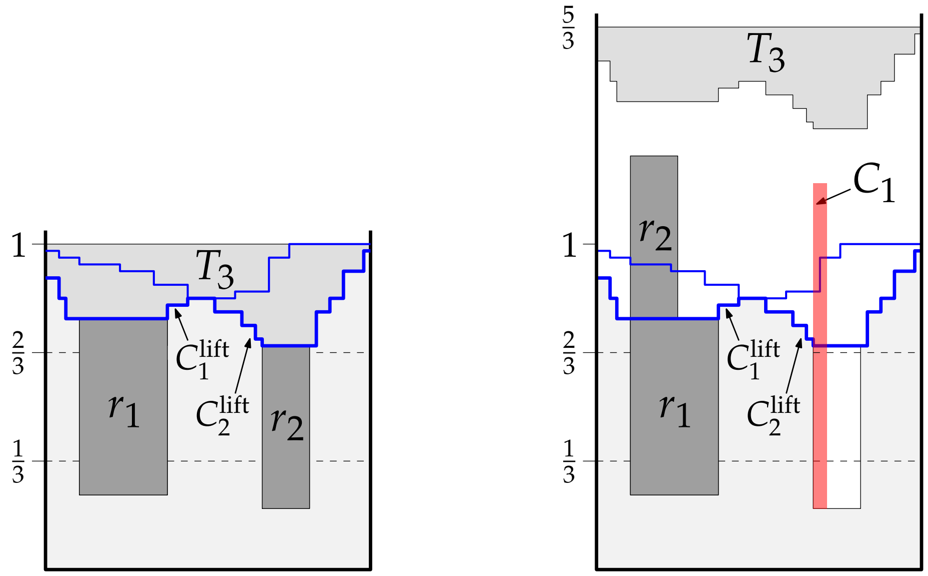
1: Define $C_1^{\mathrm{lift}} = \mathrm{VPCE}(x_1, y_1')$, $C_2^{\mathrm{lift}} = \mathrm{VPCE}(x_2, y_2')$, $T_1 = \mathrm{AI}(C_1^{\mathrm{lift}})$ and $T_2 = \mathrm{AI}(C_2^{\mathrm{lift}})$.
2: Move up $T_3 = T_1 \cup T_2$ by $2/3$.
3: Pack r_2 at postion (x_1, y_1').
4: Pack C_1 into the slot vacated by r_2 and pack C_2 above the entire packing.

3.2.9 GAP BETWEEN INNERMOST 2/3-HIGH EDGES

The pre-conditions for this section are quite technical. We first state them formally and present a motivation afterwards. Thus assume that the following conditions on P are satisfied throughout this section for some small constant c_3 (think of $c_3 = 2$ for most cases).

14.1. There are rectangles $r_\ell, r_r \in H_{2/3}$ with x-coordinates $x_\ell' \in [4c_3\varepsilon, 1 - 4c_3\varepsilon]$ and $x_r \in [x_\ell' + 4c_3\varepsilon, 1 - 4c_3\varepsilon]$ (note that x_ℓ' refers to the right side of r_ℓ whereas x_r refers to the left side of r_r).

14.2. There is no $1/3$-high item that intersects with $[x_\ell' + (c_3 - 1)\varepsilon, x_\ell' + c_3\varepsilon] \times [0,1]$ and there is no $1/3$-high item that intersects with $[x_r - c_3\varepsilon, x_r - (c_3 - 1)\varepsilon] \times [0,1]$.



(a) Definition of T_3.

(b) Modified packing where C_1 is packed in the red slot of height 1 and width ε.

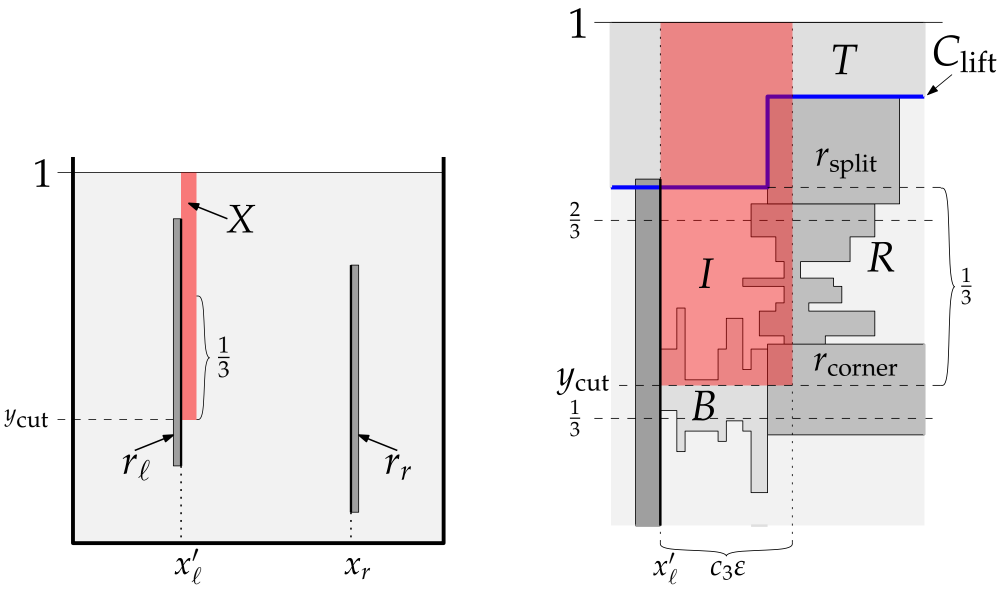
Figure 11: Packing methods for Lemma 13

To understand the motivation for these conditions assume that Lemma 9 is not applicable, which reads as follows. *If there is a* $1/3$*-high edge on the left side of the bin then the space to the left of this edge is almost completely occupied by* $2/3$*-high items*. Now we consider r_ℓ and r_r as the *innermost* such $2/3$-high items. By Lemma 9 we know that there are no further $1/3$-high items between x'_ℓ and x_r other than in a very thin slot next to these edges and close to the x-coordinate $1/2$ (exceptions are wide $1/3$-high items that span across these areas—these cases are handled separately). This property is captured in Condition 14.2. For technical reasons we require $x_r \geq x'_\ell + 4c_3\varepsilon$. If this is not the case (and Lemma 9 is not applicable), we have $w(H_{2/3}) \geq 1 - 6c_3\varepsilon$ and can apply Lemma 8.

BASIC ALGORITHM. We are going to cut out a certain slot of width $c_3\varepsilon$ next to x'_ℓ. The depth of this slot depends on the particular packing P. In a first step we describe our basic algorithm and assume that we cut down to height $y_{\text{cut}} \in [1/3, 2/3]$. In a second step we show how this basic algorithm is used depending on P and prove that it actually returns a valid packing.

Let $X = [x'_\ell, x'_\ell + c_3\varepsilon] \times [y_{\text{cut}}, 1]$ be the designated slot that we want to free. To do this we differentiate five sets of items that intersect X. The definition of these sets depends on the item $r_{\text{corner}} = \text{PointI}(x'_\ell + c_3\varepsilon, y_{\text{cut}})$ and on the item $r_{\text{split}} = \text{PointI}(x'_\ell + c_3\varepsilon, y_{\text{cut}} + 1/3)$, which are the items that reach into X from the right at height y_{cut} and $y_{\text{cut}} + 1/3$, respectively. If no item contains $(x'_\ell + c_3\varepsilon, y_{\text{cut}})$ or no item contains $(x'_\ell + c_3\varepsilon, y_{\text{cut}} + 1/3)$ in its interior, we introduce dummy items of size $(0, 0)$ for r_{corner} and r_{split}, respectively.

One further important ingredient of the basic algorithm (or rather its correctness) is the following *blocking property*. No item that intersects the designated slot X, i.e., that



(a) The items r_ℓ and r_r enforce the blocking property in the bottom of X.

(b) Distorted close-up on all sets except $X_{1/3}$ (which consists of $1/3$-high items than can occur between x'_ℓ and $x'_\ell + (c_3 - 1)\varepsilon$) for $y_{\text{cut}} = 1/3$. I consists of all items that lie completely inside X (the shaded region), T consists of all items that lie completely above C_{lift}, items in B reach into X from below, items in R reach into X from the right.

Figure 12: Blocking property and definition of sets that intersect X.

intersects $[x'_\ell, x'_\ell + c_3\varepsilon] \times [y_{\text{cut}}, y_{\text{cut}} + 1/3]$, reaches to the left of x'_ℓ or to the right of $x_r + c_3\varepsilon$, i.e., $\text{VLI}(x'_\ell; y_{\text{cut}}, y_{\text{cut}} + 1/3) = \emptyset$ and $\text{VLI}(x'_\ell + c_3\varepsilon; y_{\text{cut}}, y_{\text{cut}} + 1/3) \cap \text{VLI}(x_r + c_3\varepsilon; y_{\text{cut}}, y_{\text{cut}} + 1/3) = \emptyset$.

Intuitively, we think of r_ℓ and r_r as the blocking items, i.e., $y_\ell, y_r \leq y_{\text{cut}}$ and $y'_\ell, y'_r \geq y_{\text{cut}} + 1/3$. Hence no item intersecting X can reach beyond r_ℓ and r_r. In one special case, we cannot ensure that r_r is such a blocking item, i.e., $y_{\text{cut}} + 1/3 > y'_r$. In this case we need the additional area of width $c_3\varepsilon$ to the right of x_r. See Figure 12a for an illustration of this property, Figure 12b for an illustration of the different sets of items that intersect X and Figure 13 for an illustration of the following basic algorithm.

For the moment we assume that the blocking property is satisfied for y_{cut} and thus no item reaches into X from the left below $y_{\text{cut}} + 1/3$.

- Let $X_{1/3} = \text{AI}(x'_\ell, x'_\ell + c_3\varepsilon; 0, 1) \cap H_{1/3}$ be the set of $1/3$-high items that lie completely within $c_3\varepsilon$ distance to the right of x'_ℓ. By Condition 14.2, the total width that the items of $X_{1/3}$ occupy in the packing is bounded by $(c_3 - 1)\varepsilon$. Therefore, we can remove these items and pack them into a container $C_{X_{1/3}} = ((c_3 - 1)\varepsilon, 1)$ by preserving the packing of the rectangles in $X_{1/3}$. We put $C_{X_{1/3}}$ aside for later insertion into the free slot X together with C_1.

- Let $B = \text{HLI}(x'_\ell, x_{\text{corner}}; y_{\text{cut}}) \setminus (X_{1/3} \cup \{r_{\text{corner}}\})$ be the set of remaining items that intersect the *bottom* of X excluding r_{corner}. Pack B bottom-aligned into a container $C_B = (c_3\varepsilon, 1/3)$. This is possible since $x_{\text{corner}} - x'_\ell < c_3\varepsilon$ and the $1/3$-

high items have been removed before. Place this container at the left side of the strip above the current packing at position $(0,1)$.

- Let $I = \mathrm{AI}(X) \setminus X_{1/3}$ be the set of remaining items that lie completely *inside* of X. There are no $1/3$-high items in I due to the removal of $X_{1/3}$ but the packing has a total height of $1 - y_{\mathrm{cut}} \in [1/3, 2/3]$. We use a standard method to repack I into a container of height $1/3$ as follows. Let $I_1 \subseteq I$ be the subset of items that intersect height $y = 2/3$ (these items can be bottom-aligned to fit into $(c_3\varepsilon, 1/3)$) and let $I_2 \subseteq I$ and $I_3 \subseteq I$ be the subsets of I that lie completely above or below $y = 2/3$, respectively. By preserving the packing of I_2 and I_3 we can pack I into $C_I = (3c_3\varepsilon, 1/3)$. Place this container next to C_B at position $(c_3\varepsilon, 1)$. The container does not intersect with the space above the designated slot X since the combined width of C_B and C_I is $4c_3\varepsilon$ and $x'_\ell \geq 4c_3\varepsilon$ by Condition 14.1.

- Consider the contour C_{lift} defined by height $y = y_{\mathrm{cut}} + 1/3$ to the left of r_{split} and by the top of r_{split} to the right. More formally, let

  $$C_{\mathrm{lift}} = \mathrm{PL}\big((0, y_{\mathrm{cut}} + 1/3), (x_{\mathrm{split}}, y_{\mathrm{cut}} + 1/3), (x_{\mathrm{split}}, y'_{\mathrm{split}}), (1, y'_{\mathrm{split}})\big).$$

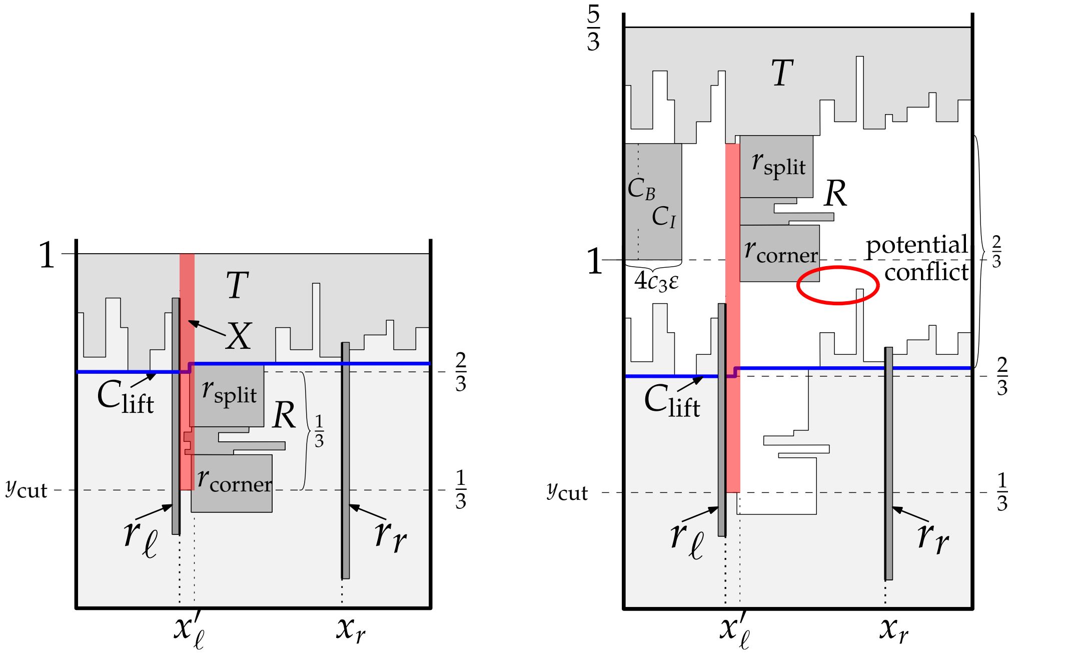
  Let $T = \mathrm{AI}(C_{\mathrm{lift}}) \setminus I$ be the set of items that lie completely above this contour but not in I. Move T up by $2/3$. This does not cause an overlap with the containers C_B and C_I since C_{lift} lies completely above $2/3$ (as $y_{\mathrm{cut}} \geq 1/3$) and thus the lowest item in T reaches a final position above $4/3$.

- Let $R = \mathrm{VLI}(x'_\ell + c_3\varepsilon; y_{\mathrm{cut}}, y_{\mathrm{cut}} + 1/3)$ be the set of items that intersect with the *right* side of X up to the crucial height of $y_{\mathrm{cut}} + 1/3$. Move R up by $2/3$ and then left-align all items at x-coordinate $x'_\ell + c_3\varepsilon$. This is the sole operation in the basic algorithm that might cause a conflict. This potential conflict only affects r_{corner} and we will later see how to overcome this difficulty. All other items were entirely above y_{cut} in the original packing and are thus moved above height 1. Therefore, they cannot overlap with any item inside the original packing P. Since the blocking property ensures that no item of R has width greater than $x_r + c_3\varepsilon - x'_\ell$ and $x_r \leq 1 - 4c_3\varepsilon$ (by Condition 14.1) we can left-align all items at x-coordinate $x'_\ell + c_3\varepsilon$ without any item intersecting the right side of the strip.

Finally, after resolving the potential conflict from the last step, we insert C_1 and $C_{X_{1/3}}$ into the slot X at position $(x'_\ell, y_{\mathrm{cut}})$ (they fit since $w(C_1) + w(C_{X_{1/3}}) \leq c_3\varepsilon$ and all items in T lie above $y_{\mathrm{cut}} + 1/3 + 2/3 = y_{\mathrm{cut}} + 1$) and pack C_2 above the entire packing as always. See Algorithm 6 for the complete basic algorithm.

We described the basic algorithm in a way that it always cuts down from the top of the packing next to r_ℓ. But there are four potential cuts since we can also cut next to r_r or from below. To ease the presentation we will stick to cutting next to r_ℓ from above by otherwise mirroring the packing horizontally and/or vertically.

Now let us see how to invoke the basic algorithm such that the blocking property is satisfied for y_{cut} and how to resolve the potential conflict.



(a) Starting situation with $y_{\text{cut}} = 1/3$.

(b) Generated packing where C_1 and $C_{X_{1/3}}$ are packed into the red slot of height 1 and width $c_3\varepsilon$. Since r_{corner} reaches below 1 in this case, a conflict might occur between r_{corner} and an item that lies to the right of r_{split} and reaches into T. In the case shown here, r_{corner} is moved to the right of r_{split} to resolve the potential conflict.

Figure 13: The basic algorithm

Algorithm 6 Basic algorithm

Requirement: Packing P that satisfies Conditions 14.1 and 14.2; $y_{\text{cut}} \in [1/3, 2/3]$ that satisfies the blocking property

Remove the items $X_{1/3} = \text{AI}(x'_\ell, x'_\ell + c_3\varepsilon; 0, 1) \cap H_{1/3}$ and pack them into a container $C_{X_{1/3}} = ((c_3 - 1)\varepsilon, 1)$.

Pack $B = \text{HLI}(x'_\ell, x_{\text{corner}}; y_{\text{cut}}) \setminus (X_{1/3} \cup \{r_{\text{corner}}\})$ into a container $C_B = (c_3\varepsilon, 1/3)$ at position $(0, 1)$.

Pack $I = \text{AI}(X) \setminus X_{1/3}$ into a container $C_I = (3c_3\varepsilon, 1/3)$ at position $(c_3\varepsilon, 1)$.

Move up the items $T = \text{AI}(C_{\text{lift}}) \setminus I$ by $2/3$.

Move up the items $R = \text{VLI}(x'_\ell + c_3\varepsilon; y_{\text{cut}}, y_{\text{cut}} + 1/3)$ by $2/3$ and left-align them with $x'_\ell + c_3\varepsilon$.

Resolve potential conflicts.

Pack C_1 and $C_{X_{1/3}}$ into the slot X at position $(x'_\ell, y_{\text{cut}})$ and pack C_2 above the entire packing.

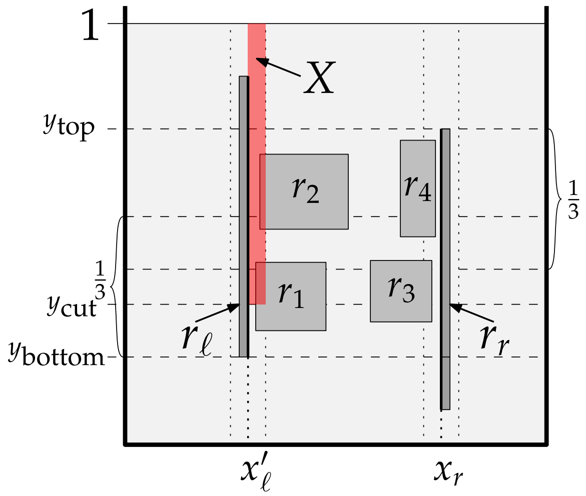


Figure 14: Definition of y_{top}, y_{bottom}, r_1, r_2, r_3 and r_4 and accentuation of the designated slot X next to r_ℓ. Here r_1 intersects with $1/3$ and $w_1 + w_2 \leq 1 - x'_\ell - c_3\varepsilon$ and thus $y_{cut} = 1/3$.

Let $y_{top} = \min(y'_\ell, y'_r) > 2/3$ and let $y_{bottom} = \max(y_\ell, y_r) < 1/3$. We get the items

$$r_1 = \mathrm{PointI}(x'_\ell + c_3\varepsilon, y_{top} - 1/3),$$
$$r_2 = \mathrm{PointI}(x'_\ell + c_3\varepsilon, y_{bottom} + 1/3),$$
$$r_3 = \mathrm{PointI}(x_r - c_3\varepsilon, y_{top} - 1/3), \text{ and}$$
$$r_4 = \mathrm{PointI}(x_r - c_3\varepsilon, y_{bottom} + 1/3)$$

as some potential corner pieces of the cut (see Figure 14), corresponding to item r_{corner} in the basic algorithm. Note that the items r_1, r_2, r_3 and r_4 do not necessarily have to differ or to exist (while in the latter case we again introduce a dummy item of size $(0,0)$) . By Condition 14.2 we know that $h_1, h_2, h_3, h_4 \leq 1/3$.

In the following cases we set $y_{cut} \in [1/3, y_{top} - 1/3]$. For this value of y_{cut} the blocking property is satisfied since $y_\ell, y_r \leq 1/3 \leq y_{cut}$ and $y'_\ell, y'_r \geq y_{top} \geq y_{cut} + 1/3$. Thus $\mathrm{VLI}(x'_\ell; y_{cut}, y_{cut} + 1/3) \subseteq \mathrm{VLI}(x'_\ell; y_\ell, y'_\ell) = \emptyset$ and $\mathrm{VLI}(x'_\ell + c_3\varepsilon; y_{cut}, y_{cut} + 1/3) \cap \mathrm{VLI}(x_r + c_3\varepsilon; y_{cut}, y_{cut} + 1/3) \subseteq \mathrm{VLI}(x'_\ell + c_3\varepsilon; y_{cut}, y_{cut} + 1/3) \cap \mathrm{VLI}(x_r; y_r, y'_r) \subseteq \mathrm{VLI}(x_r; y_r, y'_r) = \emptyset$ (no item from $[x'_\ell, x'_\ell + c_3\varepsilon] \times [y_{cut}, y_{cut} + 1/3]$ reaches beyond x'_ℓ and x_r).

We now describe the different cases in which we invoke the basic algorithm.

CASE 1. $y_1 \geq 1/3$, i.e., r_1 lies above height $1/3$.
In this case we invoke the basic algorithm with $y_{cut} = y_1 \in [1/3, y_{top} - 1/3]$ (hence the blocking property is satisfied). We have $r_{corner} = (0,0)$ and r_1 is the lowest item in R. The item r_1 is moved above height 1 since $y_1 \geq 1/3$. Thus no conflict occurs.

In the following assume conversely that $y_1, y_3 < 1/3$ and $y'_2, y'_4 > 2/3$ (using mirroring). This implies that r_1, r_3 intersect the horizontal line at height $y = 1/3$ and r_2, r_4 intersect the horizontal line at height $y = 2/3$ (by definition of the potential corner pieces and as $y_{top} - 1/3 > 1/3$ and $y_{bottom} + 1/3 < 2/3$). Hence, we have $r_1 \neq r_2$ and $r_3 \neq r_4$ since $h_1, h_2, h_3, h_4 \leq 1/3$.

CASE 2. $w_1 + w_2 \leq 1 - x'_\ell - c_3\varepsilon$ (and r_1 intersect $y = 1/3$, r_2 intersects $y = 2/3$).
In this case we potentially mirror the packing horizontally, i.e., over $y = 1/2$, to ensure that $h_1 \leq h_2$. We invoke the basic algorithm with $y_{\text{cut}} = 1/3$ (hence the blocking property is satisfied) and thus we have $r_{\text{corner}} = r_1$ and $r_{\text{split}} = r_2$ (as we can assume from the previous case that r_1 intersects $y = 1/3$ and r_2 intersects $y = 2/3$). Since r_1 intersects the horizontal line $y = 1/3$ it is not moved out of the original packing P (and could therefore cause a conflict with the original packing). Since $w_1 + w_2 \leq 1 - x'_\ell - c_3\varepsilon$ and $h_1 \leq h_2$ we can pack r_1 to the right of r_2 which is left-aligned at x-coordinate $x'_\ell + c_3\varepsilon$, i.e., pack r_1 at position $(x'_\ell + c_3\varepsilon + w_2, y_2 + 2/3)$. This handles the potential conflict of the basic algorithm.

In the following we assume that $w_1 + w_2 > 1 - x'_\ell - c_3\varepsilon$ and accordingly $w_3 + w_4 > x_r - c_3\varepsilon$. Thus $\sum_{i=1}^{4} w_i > 1 + x_r - x'_\ell - 2c_3\varepsilon > 2(x_r - x'_\ell)$. This is obviously only possible if $r_1 = r_3$ or $r_2 = r_4$. Let us thus assume $r_2 = r_4$ (by otherwise mirroring the packing over $y = 1/2$).

CASE 3. $w_1 \leq x_r - x'_\ell - 2c_3\varepsilon$ (and $r_2 = r_4$ and r_1 intersects $y = 1/3$ and r_2 intersects $y = 2/3$).
Again we invoke the basic algorithm with $y_{\text{cut}} = 1/3$ (hence the blocking property is satisfied) and accordingly we have $r_{\text{corner}} = r_1$ and $r_{\text{split}} = r_2$. All items above r_2 are in T, hence by moving up T by $2/3$ no item intersects the area above r_2, that is, in particular, the area $[x'_\ell + c_3\varepsilon, x_r - c_3\varepsilon] \times [y'_2, 1]$. Furthermore, since r_2 intersects the horizontal line $y = 2/3$, except r_1 no item is placed in $[x'_\ell + c_3\varepsilon, x_r - c_3\varepsilon] \times [2/3, 1]$ after moving up R by $2/3$. The rectangle r_1 has height at most $1/3$ and width at most $x_r - x'_\ell - 2c_3\varepsilon$. Hence by moving up r_1 by $2/3$ and left-aligning it at x-coordinate $x'_\ell + c_3\varepsilon$ it intersects only with the free area $[x'_\ell + c_3\varepsilon, x_r - c_3\varepsilon] \times [2/3, 1]$ inside the original packing P. Thus no conflict occurs.

On the other hand, if conversely $w_1 > x_r - x'_\ell - 2c_3\varepsilon$ and accordingly $w_3 > x_r - x'_\ell - 2c_3\varepsilon$ we have $r_1 = r_3$ since $x_r \geq x'_\ell + 4c_3\varepsilon$ by Condition 14.1.

Thus for the last case we have $r_1 = r_3$ and $r_2 = r_4$ and r_1 intersects height $y = 1/3$ and r_2 intersects height $y = 2/3$. The challenge in this remaining case is that we cannot move r_1 out of the original packing (since it intersects $y = 1/3$) and thus there might occur a conflict close to r_r. We now show how to resolve this potential conflict close to r_r.

TWO WIDE CORNER PIECES. Let $y'_{\text{top-left}}$ be the height of the bottom of the lowest item above r_ℓ that intersects $x'_\ell - c_3\varepsilon$ and $x'_\ell + c_3\varepsilon$, i.e.,

$$y'_{\text{top-left}} = \min\left\{y_i \mid r_i \in \text{VLI}(x'_\ell - c_3\varepsilon; y'_\ell, 1) \cap \text{VLI}(x'_\ell + c_3\varepsilon; y'_\ell, 1)\right\}.$$

If there is no such item let $y'_{\text{top-left}} = 1$. Let $y'_{\text{top-right}}$, $y'_{\text{bottom-left}}$ and $y'_{\text{bottom-right}}$ be defined accordingly as shown in Figure 15. Now we define $y'_{\text{top}} = \min(y'_{\text{top-left}}, y'_{\text{top-right}})$ and $y'_{\text{bottom}} = \max(y'_{\text{bottom-left}}, y'_{\text{bottom-right}})$. Let us assume that $y'_{\text{top}} = y'_{\text{top-right}}$ (by otherwise mirroring over $x = 1/2$).

CASE 4. $y_1 \geq y'_{\text{top}} - 2/3$ (and $r_1 = r_3$ and $r_2 = r_4$ and r_1 intersects $y = 1/3$ and r_2 intersects $y = 2/3$).

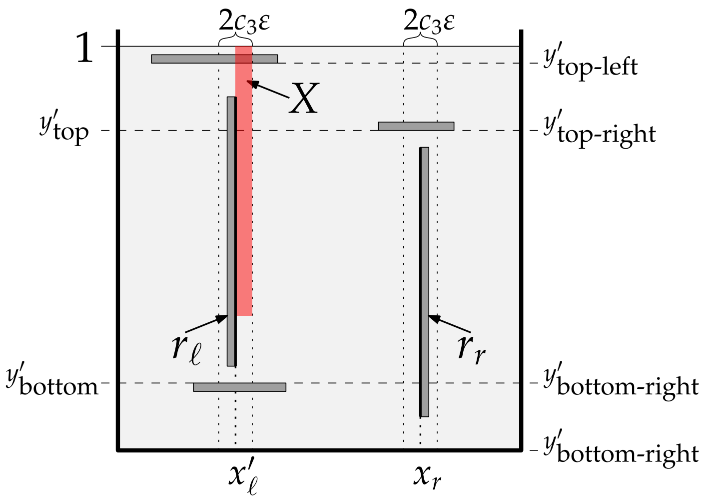


Figure 15: Definition of y'_{top} and y'_{bottom}. No item from X below y'_{top} reaches beyond $x_r + c_3\varepsilon$.

In this case we invoke the basic algorithm with $y_{\text{cut}} = 1/3$ as usual (hence the blocking property is satisfied) and again we have $r_{\text{corner}} = r_1$ and $r_{\text{split}} = r_2$. Let r_5 be the item that defined $y'_{\text{top}} = y'_{\text{top-right}} = y_5$. Then the items r_2 and r_5 intersect the vertical line $x = x_r - c_3\varepsilon$. Since $y'_{\text{top}} \geq \min(y'_\ell, y'_r) > 2/3$ and r_2 intersects the horizontal line $y = 2/3$ and since r_2 and r_5 intersect the same vertical line, it follows that $y_5 = y'_{\text{top}} \geq y'_2$. So r_5 and all rectangles above r_5 are in T and moved up by $2/3$. Therefore, no item intersects with the area $[x'_\ell + c_3\varepsilon, x_r + c_3\varepsilon] \times [y'_{\text{top}}, 1]$. Since $y_1 \geq y'_{\text{top}} - 2/3$, we move up r_1 above y'_{top} and into this area. Thus no conflict occurs. So in the following assume conversely that $y_1 < y'_{\text{top}} - 2/3$ and accordingly $y'_2 > y'_{\text{bottom}} + 2/3$.

CASE 5. $y_1 < y'_{\text{top}} - 2/3$ and $y'_2 > y'_{\text{bottom}} + 2/3$.
Now assume that $y_{\text{bottom}} = y_\ell$ (by otherwise mirroring vertically—so $y'_{\text{top}} = y'_{\text{top-right}}$ does not necessarily hold any more). Note that we refer to the original definition of y_{bottom} instead of y'_{bottom} here. We invoke the basic algorithm using $y_{\text{cut}} = y'_1$. So r_1 is left in the original position (and we have $r_{\text{corner}} = (0,0)$) and since $y'_1 > 1/3$ all items from R are moved out of the original packing. It remains to verify the blocking property for $y_{\text{cut}} = y'_1$, since y'_1 is not necessarily in $[1/3, y_{\text{top}} - 1/3]$.

We have $y'_\ell > y_\ell + 2/3 = y_{\text{bottom}} + 2/3 \geq y'_1 + 1/3 = y_{\text{cut}} + 1/3$ (since $y'_1 \leq y_{\text{bottom}} + 1/3$ as by definition r_2 intersects with $y_{\text{bottom}} + 1/3$ and $r_1 \neq r_2$). So the blocking property is enforced by r_ℓ to the left, i.e., $\text{VLI}(x'_\ell; y_{\text{cut}}, y_{\text{cut}} + 1/3) = \emptyset$. Moreover, we have $y_{\text{cut}} + 1/3 < y'_{\text{top}}$ since $y_{\text{cut}} + 1/3 = y'_1 + 1/3 = y_1 + h_1 + 1/3 \leq y_1 + 2/3 < y'_{\text{top}}$. Thus by definition of y'_{top} no item that intersects $x_r - c_3\varepsilon$ between y'_r and y'_{top} reaches beyond $x_r + c_3\varepsilon$, i.e., $\text{VLI}(x'_\ell + c_3\varepsilon; y_{\text{cut}}, y_{\text{cut}} + 1/3) \cap \text{VLI}(x_r + c_3\varepsilon; y_{\text{cut}}, y_{\text{cut}} + 1/3) = \emptyset$. So the blocking property is also satisfied for the right side.

In total we get the following lemma.

Lemma 14. *Let $c_3 > 0$ be a constant. If the following conditions hold for P, namely*

14.1. there are rectangles $r_\ell, r_r \in H_{2/3}$ with x-coordinates $x'_\ell \in [4c_3\varepsilon, 1 - 4c_3\varepsilon]$ and $x_r \in [x'_\ell + 4c_3\varepsilon, 1 - 4c_3\varepsilon]$, and

14.2. *there is no* $1/3$*-high item that intersects with* $[x'_\ell + (c_3 - 1)\varepsilon, x'_\ell + c_3\varepsilon] \times [0,1]$ *and there is no* $1/3$*-high item that intersects with* $[x_r - c_3\varepsilon, x_r - (c_3 - 1)\varepsilon] \times [0,1]$*,*

then we can derive a packing of I *into a strip of height* $5/3 + \varepsilon$ *in additional time* $\mathcal{O}(n \log n)$*.*

We use the same methods, namely the basic algorithm invoked with $y_{\text{cut}} = 1/3$ and $c_3 = 2$ for another case where we do not have a blocking edge of height $2/3$ on both sides. More specifically, we get the following corollary where the right-hand blocking item r_r is $1/3$-high.

Corollary 4. *If the following conditions hold for* P*, namely*

14.3. *there is an item* $r_\ell \in H_{2/3}$ *with x-coordinate* $x'_\ell \in [8\varepsilon, 1/2 - 9\varepsilon]$*, and*

14.4. *there is an item* $r_r \in H_{1/3}$ *that intersects* $y = 1/3$ *and* $y = 2/3$ *with x-coordinate* $x_r \in [1/2 - \varepsilon, 1/2 + \varepsilon]$*, and*

14.5. *there is no* $1/3$*-high item that intersects with* $[x'_\ell + \varepsilon, x'_\ell + 2\varepsilon] \times [0,1]$*,*

then we can derive a packing of I *into a strip of height* $5/3 + \varepsilon$ *in additional time* $\mathcal{O}(n \log n)$*.*

Proof. As stated above, we invoke the basic algorithm with $y_{\text{cut}} = 1/3$ and $c_3 = 2$. Note that $x_r \geq x'_\ell + 8\varepsilon = x'_\ell + 4c_3\varepsilon$. The blocking property is satisfied, since r_ℓ and r_r intersects with the horizontal lines at height $y = 1/3$ and $y = 2/3$. If $r_{\text{corner}} = (0,0)$ or $r_{\text{split}} = (0,0)$ we can use the same methods as in Case 1. Otherwise r_{corner} intersects $y = 1/3$ and r_{split} intersects $y = 2/3$. Since r_r also intersects $y = 1/3$ and $y = 2/3$ we have $w_{\text{corner}} \leq x_r - x'_\ell$ and $w_{\text{split}} \leq x_r - x'_\ell$. Thus $w_{\text{corner}} + w_{\text{split}} \leq 2x_r - 2x'_\ell \leq 1 + 2\varepsilon - 2x'_\ell < 1 - x'_\ell - 4\varepsilon = 1 - x'_\ell - 2c_3\varepsilon$. Hence we can use the same methods as in Case 2. □

3.2.10 THE OVERALL ALGORITHM

In this section we finally bring together the methods described in the previous sections to give the overall algorithm.

Recall that $\varepsilon < 1/(28 \cdot 151)$. We start by applying the methods of Lemma 3 in order to reduce the problem to pack an instance I with $\text{OPT}(I) \leq 1$. In a first step we attempt to apply Lemma 7 or Lemma 8. If we are successful, this gives a packing into a strip of height at most $5/3 + 260\varepsilon/3$ which we return. Otherwise we have

$$h(W_{1-130\varepsilon}) < 1/3 \quad \text{and} \tag{3.7}$$

$$w(H_{2/3}) < 27/28. \tag{3.8}$$

Now we apply the $\mathcal{PTAS}$ from [5] with an accuracy of $\delta := \varepsilon^2/2$ and denote the resulting packing by P. We pack the remaining items into $C_1 = (\varepsilon, 1)$ and $C_2 = (1, \varepsilon)$ as described in Section 3.2.1. According to P we decide which of the other Lemmas to apply in order to insert C_1 into a free slot in P.

The $2/3$-high items play a crucial role in this decision. Let r_ℓ be the rightmost $2/3$-high item in $\text{AI}(0, 1/2 - \varepsilon; 0, 1)$ and let r_r be the leftmost $2/3$-high item in $\text{AI}(1/2 +$

$\varepsilon, 1; 0, 1)$ (we consider the sides of the strip as items of height 1 and width 0 to ensure that r_ℓ and r_r exist).

Iterate over all 1/3-high items in P and check the applicability of Lemma 9 for both sides of each item (also under mirroring over $x = 1/2$). If possible, use the methods of Lemma 9 to derive a packing into a strip of height $5/3 + \varepsilon$. Otherwise P has the following properties.

- The areas to the left of r_ℓ and to the right of r_r are almost completely covered by 2/3-high items, i.e., $w(\mathrm{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) > x'_\ell - \varepsilon$ and $w(\mathrm{AI}(x_r, 1; 0, 1) \cap H_{2/3}) > 1 - x_r - \varepsilon$, since Lemma 9 could not be applied on r_ℓ and r_r.
- The x-coordinates of the sides of all 1/3-high items are in $I_\ell = [0, x'_\ell + \varepsilon]$, $I_M = [1/2 - \varepsilon, 1/2 + \varepsilon]$ or $I_r = [x_r - \varepsilon, 1]$. To put it in another way the rectangles in $H_{1/3}$ are either completely inside one of these intervals or span across one interval to another.
- We have $x_r - x'_\ell > 143\varepsilon$ as otherwise

$$\begin{aligned} w(H_{2/3}) &\geq w(\mathrm{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) + w(\mathrm{AI}(x_r, 1; 0, 1) \cap H_{2/3}) \\ &\geq x'_\ell - \varepsilon + 1 - x_r - \varepsilon \\ &\geq 1 - 145\varepsilon \\ &> 27/28 \end{aligned}$$

  for an $\varepsilon < 1/(28 \cdot 145)$ in contradiction to Inequality (3.8).

The specific method that we apply in the next step depends on the existence of 1/3-high items that span across the intervals I_ℓ, I_M and I_r. It is not possible that a 2/3-high item r_1 spans from I_ℓ to I_r, as otherwise we have $w(H_{2/3}) \geq w(\mathrm{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) + w(\mathrm{AI}(x_r, 1; 0, 1) \cap H_{2/3}) + w_1 \geq x'_\ell - \varepsilon + 1 - x_r - \varepsilon + x_r - x'_\ell - 2\varepsilon \geq 1 - 4\varepsilon > 27/28$ for $\varepsilon < 1/112$. The same holds if there were two 2/3-high rectangles r_1, r_2, that span from I_ℓ to I_M and I_M to I_r, respectively ($w(H_{2/3}) \geq w(\mathrm{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) + w(\mathrm{AI}(x_r, 1; 0, 1) \cap H_{2/3}) + w_1 + w_2 \geq x'_\ell - \varepsilon + 1 - x_r - \varepsilon + x_r - x'_\ell - 4\varepsilon \geq 1 - 6\varepsilon > 27/28$ for $\varepsilon < 1/168$).

In order to have a consistent description of the following cases, we assume that there are no 2/3-high items that span across the intervals by otherwise redefining r_ℓ and x'_ℓ or r_r and x_r as follows. If there is a 2/3-high item r that intersects with $x = x'_\ell + \varepsilon$, i.e., r spans from I_ℓ and I_M, then we redefine r_ℓ as the rightmost 2/3-high item in I_M, or $r_\ell = r$ if there is no 2/3-high item completely in I_M. On the other hand, if there is a rectangle r that intersects with $x = x_r - \varepsilon$, i.e., r spans from I_M to I_r, then we redefine r_r as the leftmost 2/3-high item in I_M, or $r_r = r$ if no 2/3-high item is completely in I_M.

After this step we have to update the first properties above as follows.

- The areas to the left of r_ℓ and to the right of r_r are almost completely covered by 2/3-high items, i.e., $w(\mathrm{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) > x'_\ell - 4\varepsilon$ and $w(\mathrm{AI}(x_r, 1; 0, 1) \cap H_{2/3}) > 1 - x_r - 4\varepsilon$.

This property follows again from the inapplicability of Lemma 9 and the observation that only uncovered area of total width 3ε in $[x'_\ell, x'_\ell + \varepsilon]$ (for the now outdated value of x'_ℓ) and $[1/2 - \varepsilon, 1/2 + \varepsilon]$ can be added.

We still can assume $x_r - x'_\ell > 143\varepsilon$, since otherwise $w(H_{2/3}) \geq w(\text{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) + w(\text{AI}(x_r, 1; 0, 1) \cap H_{2/3}) \geq x'_\ell - 4\varepsilon + 1 - x_r - 4\varepsilon \geq 1 - 151\varepsilon \geq 27/28$ for an $\varepsilon < 1/(28 \cdot 151)$.

Let $c_3 = 2$ if $x'_\ell < 1/2 - 3\varepsilon$ and $x_r > 1/2 + 3\varepsilon$ and $c_3 = 5$ otherwise. The intention of this definition is that $[x'_\ell + (c_3 - 1)\varepsilon, x'_\ell + c_3\varepsilon]$ does not intersect with $I_\ell \cup I_M$ and $[x_r - c_3\varepsilon, x_r - (c_3 - 1)\varepsilon]$ does not intersect with $I_M \cup I_r$ (here we use $x_r - x'_\ell > 143\varepsilon$ as thus if I_ℓ lies close to I_M we have a bigger gap between I_M and I_r and vice versa). Since the x-coordinates of the sides of all 1/3-high items are in I_ℓ, I_M and I_r we thus get the following property for P.

- If one 1/3-high item intersects with $[x'_\ell + (c_3 - 1)\varepsilon, x'_\ell + c_3\varepsilon] \times [0, 1]$, then it has to cross the vertical line at $x = x'_\ell + c_3\varepsilon$.
- If one 1/3-high item intersects with $[x_r - c_3\varepsilon, x_r - (c_3 - 1)\varepsilon] \times [0, 1]$, then it has to cross the vertical line at $x = x_r - c_3\varepsilon$.

Now assume that $x'_\ell \leq 4c_3\varepsilon$ and no 1/3-high item intersects with $x = x'_\ell + c_3\varepsilon$. Thus no 1/3-high item spans across I_ℓ and I_M and the precondition of Lemma 10 with $c_1 = 5c_3$ is satisfied (we have $h(W_{1-5(c_1+1)\varepsilon}) = h(W_{1-5(5c_3+1)\varepsilon}) \leq h(W_{1-130\varepsilon}) < 1/3$ by Condition 3.7). We use the methods of Lemma 10 to derive a packing into a strip of height $5/3 + \varepsilon$ which we return. For a packing P that is still not processed we get the following property.

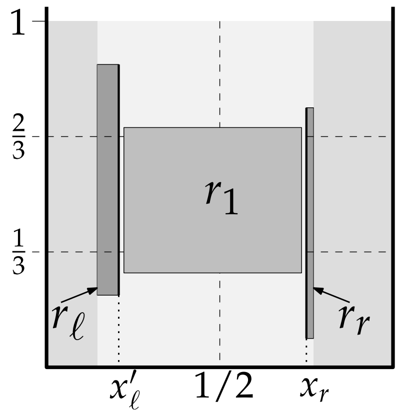
- If no 1/3-high item intersects with $x = x'_\ell + c_3\varepsilon$, then $x'_\ell \geq 4c_3\varepsilon$ and analogously if no 1/3-high item intersects with $x = x_r - c_3\varepsilon$, then $x_r \leq 1 - 4c_3\varepsilon$.

See Figure 16 for a schematic illustration of the following cases (by the considerations above, all 1/3-high items that span across the intervals have height at most 2/3).

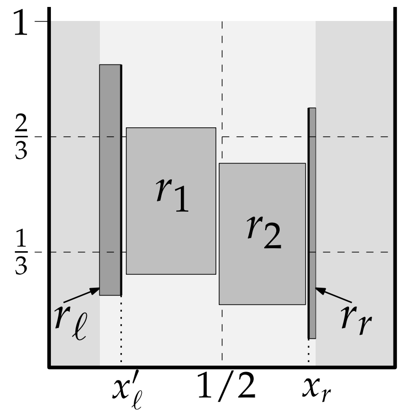
- *A 1/3-high item reaches close to r_ℓ and r_r—see Figure 16a.*

  In this case we assume that there is a 1/3-high item r_1 that intersects with $x = x'_\ell + \varepsilon$ and with $x = x_r - \varepsilon$, i.e., that spans from I_ℓ to I_r. By Inequality (3.7) we have $w_1 \leq 1 - 130\varepsilon$ as $h_1 > 1/3$. Moreover, we have $w_1 \geq x_r - \varepsilon - x'_\ell - \varepsilon \geq 141\varepsilon$ (since $x_r - x'_\ell > 143\varepsilon$). Thus if $y_1 \geq 1/3$ or $y'_1 \leq 2/3$ we can apply the methods of Lemma 11. Otherwise, we can apply Algorithm 4 with $c_2 = 10$ since $w_1 \geq x_r - \varepsilon - x'_\ell - \varepsilon \geq 141\varepsilon > (4c_2 + 1)\varepsilon$ (since $x_r - x'_\ell > 143\varepsilon$) and $w(H_{2/3}) \geq w(\text{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) + w(\text{AI}(x_r, 1; 0, 1) \cap H_{2/3}) \geq x'_\ell - 4\varepsilon + 1 - x_r - 4\varepsilon \geq 1 - w_1 - 10\varepsilon = 1 - w_1 - c_2\varepsilon$.

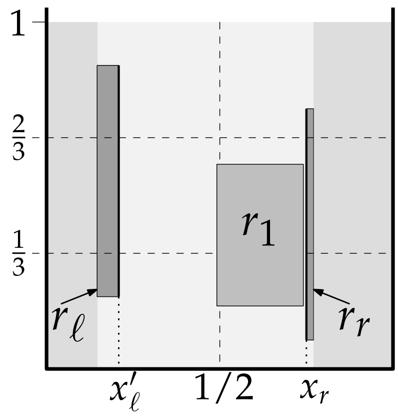
  In the following we also need to handle the case where r_1 reaches only close to the blocking items r_ℓ and r_r, i.e., r_1 intersects with $x'_\ell + 11\varepsilon$ and $x_r - 11\varepsilon$. Here we can also apply the methods of Lemma 11 or Algorithm 4 with $c_2 = 30$ ($w_1 \geq x_r - 11\varepsilon - x'_\ell - 11\varepsilon \geq 121\varepsilon = (4c_2 + 1)\varepsilon$ and $w(H_{2/3}) \geq w(\text{AI}(0, x'_\ell; 0, 1) \cap H_{2/3}) + w(\text{AI}(x_r, 1; 0, 1) \cap H_{2/3}) \geq x'_\ell - 4\varepsilon + 1 - x_r - 4\varepsilon \geq 1 - w_1 - 30\varepsilon = 1 - w_1 - c_2\varepsilon$).



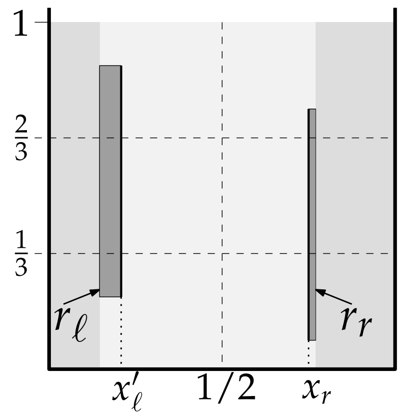
(a) A $1/3$-high item spans from I_ℓ to I_r



(b) A $1/3$-high item spans from I_ℓ to I_M and a $1/3$-high item spans from I_M to I_r



(c) A $1/3$-high item spans from I_M to I_r but no $1/3$-high spans between I_ℓ and I_M



(d) No $1/3$-high items span across the intervals

Figure 16: Schematic illustration of the remaining cases if Lemma 9 is not applicable. The area to the left of r_ℓ and the area to the right of r_r is almost completely covered by $2/3$-high items (and shown in darker shade).

- *Two* $1/3$*-high items lie between* r_ℓ *and* r_r*—see Figure 16b.*

  Assume that there is a $1/3$-high item r_1 that intersects with $x = x'_\ell + \varepsilon$ and with $x = 1/2 - \varepsilon$ and there is a $1/3$-high item r_2 that intersects with $x = 1/2 + \varepsilon$ and with $x = x_r - \varepsilon$. Note, that if r_1 or r_2 spans from I_ℓ to I_r, then we are in the previous case. Hence we assume that r_1 spans across I_ℓ to I_M and r_2 spans across I_M to I_r. If $x'_\ell \geq 1/2 - 3\varepsilon$ or $x_r \leq 1/2 + 3\varepsilon$ we apply also the method of the previous case, since then r_2 intersects with $x = x'_\ell + 5\varepsilon$ and $x = x_r - \varepsilon$, or r_1 intersects with $x = x'_\ell + \varepsilon$ and $x = x_r - 5\varepsilon$. Otherwise we have $w_1, w_2 \in [\varepsilon, 1/2 + \varepsilon]$. Thus if r_1 or r_2 does not intersect with $y = 1/3$ or with $y = 2/3$, we can apply the methods of Lemma 11. Otherwise, we have $y_1, y_2 < 1/3$ and $y'_1, y'_2 > 2/3$ and thus we can apply the methods of Lemma 13.

The following two cases use Lemma 14 and Corollary 4. Recall that we have $x'_\ell \geq 4c_3\varepsilon$ if no $1/3$-high item intersects with $x = x'_\ell + c_3\varepsilon$ and $x_r \leq 1 - 4c_3\varepsilon$ if no $1/3$-high item intersects with $x = x_r - c_3\varepsilon$.

- *A* $1/3$*-high item reaches from the middle close to* r_r *but no* $1/3$*-high item reaches from* r_ℓ *to the middle—see Figure 16c.*

  In this case we assume that there is a $1/3$-high item r_1 that intersects with $x = 1/2 + \varepsilon$ and with $x = x_r - c_3\varepsilon$ but there is no $1/3$-high item that intersects with $x = x'_\ell + c_3\varepsilon$. We assume that $x'_\ell \leq 1/2 - 3\varepsilon$ as otherwise we could apply the methods of the first case (as r_1 intersects with $x = x'_\ell + 4\varepsilon \leq x'_\ell + 11\varepsilon$ and $x = x_r - \varepsilon$ in this case). Note that we have $x_r > 1/2 + 3\varepsilon$ as otherwise $c_3 = 5$ and r_1 would intersect with $x = 1/2 - \varepsilon$, i.e., span from I_ℓ to I_r, and the assumption that no $1/3$-high intersects with $x = x'_\ell + c_3\varepsilon$ would be violated. Thus we have $c_3 = 2$ (by the definition above) and $x'_\ell \geq 8\varepsilon$.

  Obviously, we have $w_1 \in [\varepsilon, 1 - 2\varepsilon]$ and can thus use the methods of Lemma 11 if $y_1 \geq 1/3$ or $y'_1 \leq 2/3$. Otherwise, the item r_1 intersects $y = 1/3$ and $y = 2/3$. Moreover, we have $x_1 \in [1/2 - \varepsilon, 1/2 + \varepsilon]$ and thus can apply the methods of Corollary 4 to derive a packing into a strip of height $5/3 + \varepsilon$. Here we use that no $1/3$-high item intersects with $[x'_\ell + (c_3 - 1)\varepsilon, x'_\ell + c_3\varepsilon] \times [0, 1]$ and that $x'_\ell \leq 1/2 - 9\varepsilon$ since we are otherwise in the first case again (r_1 intersects with $x = x'_\ell + 11\varepsilon$ and $x = x_r - 11\varepsilon$).

  The same methods can be applied if the item r_1 reaches from r_ℓ to the middle instead.

- *No* $1/3$*-high items span across the intervals—see Figure 16d.*

  In this case we assume that no $1/3$-high item intersects with $x = x'_\ell + c_3\varepsilon$ and no $1/3$-high item intersects with $x = x_r - c_3\varepsilon$. Thus we have $x'_\ell, x_r \in [4c_3\varepsilon, 1 - 4c_3\varepsilon]$ and no $1/3$-high item intersects with $[x'_\ell + (c_3 - 1)\varepsilon, x'_\ell + c_3\varepsilon] \times [0, 1]$ and no $1/3$-high item intersects with $[x_r - c_3\varepsilon, x_r - (c_3 - 1)\varepsilon] \times [0, 1]$. As we have $x_r - x'_\ell > 143\varepsilon > 4c_3\varepsilon$ we can apply the methods of Lemma 14.

This finalizes our overall algorithm which is given in Algorithm 7. These four cases cover all possibilities and therefore our algorithm always outputs a packing into a strip of height at most $5/3 + 260\varepsilon/3$. Thus with Lemma 3 we get an approximation ratio for the overall algorithm of $5/3 + 263\varepsilon/3$. By scaling ε appropriately we proved our main theorem.

Theorem 4. *For any $\varepsilon > 0$, there is an approximation algorithm A which produces a packing of a list I of n rectangles in a strip of width 1 and height $A(I)$ such that*

$$A(I) \leq \left(\frac{5}{3} + \varepsilon\right) \mathrm{OPT}(I).$$

The running time of A is $\mathcal{O}(T_{\mathcal{PTAS}} + (n \log^2 n)/\log\log n)$, where $T_{\mathcal{PTAS}}$ is the running time of the $\mathcal{PTAS}$ from [5].

Algorithm 7 The overall algorithm

Requirement: $\varepsilon < 1/(28 \cdot 151)$.

for any scaled instance I' according to Lemma 3 **do**
  if Lemma 7 or Lemma 8 is applicable **then**
    store the packing derived by the according method.
  else
    Apply the $\mathcal{PTAS}$ from [5] with an accuracy of $\delta := \varepsilon^2/2$ and denote the resulting packing by P.
    Pack the remaining items into $C_1 = (\varepsilon, 1)$ and $C_2 = (1, \varepsilon)$.
    Iterate over all 1/3-high items in P, apply the methods of Lemma 9 if possible and store the derived packing.
    Apply the methods of Lemma 10 if $x'_\ell \leq 4c_3\varepsilon$ and no 1/3-high item intersects with $x = x'_\ell + c_3\varepsilon$ (or $x_r \geq 1 - 4c_3\varepsilon$ and no 1/3-high item intersects with $x = x_r - c_3\varepsilon$) and store the derived packing.
    Decide which of the four cases applies according to the 1/3-high items between r_ℓ and r_r, apply the methods of Lemmas 11, 12, 13, and 14 or of Corollary 4 and store the derived packing.
return the best packing that was stored

---

4 BIN PACKING WITH ROTATIONS

In this section we consider the two-dimensional bin packing problem where the items can be rotated by 90 degrees. We present a 2-approximation algorithm for this problem. This is optimal provided $\mathcal{P} \neq \mathcal{NP}$.

No previous work has been published on the absolute approximability of this problem. On the other hand, it is common knowledge that Steinberg's algorithm yields a 4-approximation algorithm for bin packing with rotations:

The following theorem was already mentioned by Jansen & Solis-Oba[27].

Theorem 5. *Steinberg's algorithm gives an absolute* 2*-approximation for strip packing with rotations.*

Proof. Rotate all items $r_i \in I$ such that $w_i \geq h_i$ and let $b := \max(2h_{\max}(I), 2\mathcal{A}(I))$. Use Steinberg's algorithm to pack I into the rectangle $(1, b)$. This is possible since $2\mathcal{A}(I) \leq b$ and $(2h_{\max}(I) - b)_+ = 0$. The claim on the approximation ratio follows from $\mathrm{OPT}(I) \geq \max(h_{\max}(I), \mathcal{A}(I)) = b/2$. □

It is well-known that a strip packing algorithm with an approximation ratio of δ directly yields a bin packing algorithm with an approximation ratio of 2δ. To see this, cut the strip packing of height h into slices of height 1 so as to get $\lceil h \rceil$ bins of the required size. The rectangles that are split between two bins can be packed into $\lfloor h \rfloor$ additional bins. The strip packing gives a lower bound for bin packing. Thus if $h \leq \delta \mathrm{OPT}_{\mathrm{strip}}(I)$, then $\lceil h \rceil + \lfloor h \rfloor \leq 2\delta \mathrm{OPT}_{\mathrm{bin}}(I)$. Accordingly, we get the following theorem.

Theorem 6. *Steinberg's algorithm yields an absolute* 4*-approximation algorithm for bin packing with rotations.*

4.1 A 2-APPROXIMATION ALGORITHM FOR BIN PACKING WITH ROTATIONS

We start our presentation by showing in Section 4.1.1 that a first approach based on an algorithm from Jansen & Solis-Oba[27] does not lead to the desired approximation ratio. Our main result is presented in Section 4.1.2. The algorithm is based on our main lemma that we prove in Section 4.1.3.

Throughout this section we assume that all items are rotated such that $w_i \geq h_i$. Therefore, Corollary 1 reads as follows here.

Corollary 5 (Jansen & Zhang[30])**.** *If the total area of a set T of items is at most* $1/2$ *and there is at most one item of height $h_i > 1/2$, then the items of T can be packed into a bin of unit size in time $\mathcal{O}((n \log^2 n)/\log\log n)$.*

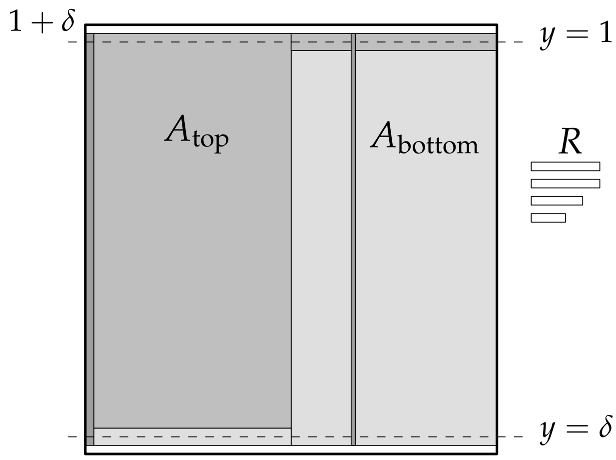


Figure 17: Packing of Jansen & Solis-Oba's algorithm where it is not immediately clear how to derive a packing into 2 unit bins. The blocks in the packing might consist of several items and might contain small free spaces or items that are not in A_{top} or A_{bottom}. Furthermore, there might be items (printed in dark grey) that are in A_{top} and in A_{bottom}.

4.1.1 FIRST APPROACH

We started our investigation on the bin packing problem with rotations with an algorithm from Jansen & Solis-Oba[27] that finds a packing of profit $(1-\delta)\text{OPT}_{2-\text{KP}}(I)$ into a bin of size $(1, 1+\delta)$, where $\text{OPT}_{2-\text{KP}}(I)$ denotes the maximal profit that can be packed into a unit bin and δ is an arbitrarily small positive constant. Using the area of the items as their profit gives an algorithm that packs almost the whole instance into an δ-augmented bin. The algorithm can easily be generalized to a constant number of bins.

An immediate idea to transform such a packing to a packing into $2\,\text{OPT}(I)$ bins is to remove all items that intersect a slice of height δ at the top or bottom of each bin. These items and the items that were not packed by the algorithm would have to be packed separately. In Figure 17 we present an instance where it is not immediately clear how the removed items can be packed separately. Let A_{top} be the set of items that intersect $y = 1$, A_{bottom} be the set of items that intersect $y = \delta$, and R be the set of remaining items. As shown in Figure 17, the sets A_{top} and A_{bottom} can both have total area arbitrary close to or even larger than $1/2$ (as both sets are not necessarily disjoint). Thus adding the additional items R and packing everything with Steinberg's algorithm is not necessarily possible. Furthermore, it is not obvious how to rearrange A_{top} or A_{bottom} such that there is suitable free space to pack R and the items that are above A_{top} or below A_{bottom}.

We therefore chose to pursue a different approach to solve this problem.

4.1.2 OUR ALGORITHM: OVERVIEW

As the asymptotic approximation ratio of the algorithm from Bansal et al.[6] is $1.525\ldots$ and thus less than 2, there exists a constant k such that for any instance with optimal value larger than k, the asymptotic algorithm gives a solution of value at most $2\,\mathrm{OPT}(I)$. This constant k is not explicitly known as we already mentioned in the introduction. We address the problem of approximating bin packing with rotations within an absolute factor of 2, provided that the optimal value of the given instance is less than k. Combined with the algorithm from [6] we get an overall algorithm with an absolute approximation ratio of 2.

We begin by applying the asymptotic algorithm from [6]. Since we do not know whether $\mathrm{OPT}(I) > k$ we apply a second algorithm that is described in the remainder of this section. If $\mathrm{OPT}(I) > k$, then this algorithm might fail as the asymptotic algorithm outputs a solution of value $k' \leq 2\,\mathrm{OPT}(I)$.

Let $\varepsilon := 1/68$. We separate the given input according to the area of the items, so we get a set of large items $L = \{r_i \in I \mid w_i h_i \geq \varepsilon\}$ and a set of small items $S = \{r_i \in I \mid w_i h_i < \varepsilon\}$. If $\mathcal{A}(L) > k$ then $\mathrm{OPT}(I) > k$ and the algorithm halts. Otherwise, we can enumerate all possible packings of the large items since the number of large items in each bin is bounded by $1/\varepsilon$ and their total area is at most k. Take an arbitrary packing of the large items into a minimum number $\ell \leq k$ of bins. If no such packing exists then the asymptotic algorithm from [6] finds a suitable solution and our algorithm halts.

If there are bins that contain items with total area less than $1/2 - \varepsilon$, we greedily add small items such that the total area of items assigned to each of these bins is in $(1/2 - \varepsilon, 1/2]$. We use the method of Jansen & Zhang[30] to repack these bins including the newly assigned small items. This is possible by Corollary 5. There is at most one item of height $h_i > 1/2$ since otherwise the total area exceeds $1/2$, because $w_i \geq h_i$. If we run out of items in this step, we found an optimal solution. Assume that there are still small items left and each bin used so far contains items of total area at least $1/2 - \varepsilon$. The following crucial lemma shows that we can pack the remaining small items well enough to achieve an absolute approximation ratio of 2.

Lemma 15. *Let $0 < \varepsilon \leq 1/68$ and let T be a set of items that all have area at most ε such that for all $r \in T$ the total area of $T \setminus \{r\}$ is less than $1/2 + \varepsilon$. We can find a packing of T into a unit bin in time $\mathcal{O}((n \log^2 n)/\log\log n)$.*

The lemma is proven in the next section. To apply Lemma 15 we consider the following partition of the remaining items.

Let $r_1, \ldots, r_m$ be the list of remaining small items, sorted by non-increasing order of size. Partition these items into sets $S_1 = \{r_{t_1}, \ldots, r_{t_2-1}\}, S_2 = \{r_{t_2}, \ldots, r_{t_3-1}\}, \ldots, S_s = \{r_{t_s}, \ldots, r_{t_{s+1}-1}\}$ with $t_1 = 1$ and $t_{s+1} = m + 1$ such that

$$\mathcal{A}(S_j \setminus \{r_{t_{j+1}-1}\}) < \frac{1}{2} + \varepsilon \qquad \text{and} \qquad \mathcal{A}(S_j) \geq \frac{1}{2} + \varepsilon$$

for $j = 1, \ldots, s-1$. Obviously, each set S_i satisfies the precondition of Lemma 15 and can therefore be packed into a single bin. Only S_s might have total area less than $1/2 + \varepsilon$. The overall algorithm is given in Algorithm 8.

Note that if no packing of L into at most k bins exists, then $\mathrm{OPT}(I) > k$ and thus $k' \leq 2\,\mathrm{OPT}(I)$ by definition of k.

Algorithm 8 Approximate bin packing with rotations

```
Apply the asymptotic algorithm from [6] to derive a packing P' into k' bins
Let ε = 1/68
Partition I into L = {r_i ∈ I | w_i h_i ≥ ε} and S = {r_i ∈ I | w_i h_i < ε}
if A(L) > k or L cannot be packed in k or less bins then
    return P'
else
    Find a packing of L into ℓ ≤ k bins, where ℓ is minimal
    while there exists a bin containing items of total area < 1/2 − ε do
        Assign small items to this bin until the total area exceeds 1/2 − ε
        Use Steinberg's algorithm (Corollary 5) to repack the bin
    Order the remaining small items by non-increasing size
    Greedily partition the remaining items into sets S_1, ..., S_s such that
        A(S_j \ {r_{t_{j+1}-1}}) < 1/2 + ε   and   A(S_j) ≥ 1/2 + ε   for j = 1, ..., s − 1
    Use the method described in the proof of Lemma 15 to pack each set S_i into
    a bin
    Let P be the resulting packing into ℓ + s bins
return the packing from P, P' that uses the least amount of bins
```

4.1.3 PACKING SETS OF SMALL ITEMS

In this section we prove Lemma 15. We will use the following partition of a set T of items of area at most ε in the remainder of this section. Let

$$T_1 := \{r_i \in T \mid 2/3 < w_i\} \qquad T_2 := \{r_i \in T \mid 1/2 < w_i \leq 2/3\}$$
$$T_3 := \{r_i \in T \mid 1/3 < w_i \leq 1/2\} \qquad T_4 := \{r_i \in T \mid w_i \leq 1/3\}.$$

Since $w_i h_i \leq \varepsilon$ and $w_i \geq h_i$, the heights of the items in each set are bounded as follows.

$$h_i \leq 3/2 \cdot \varepsilon \quad \text{for } r_i \in T_1, \qquad h_i \leq 2 \cdot \varepsilon \quad \text{for } r_i \in T_2,$$
$$h_i \leq 3 \cdot \varepsilon \quad \text{for } r_i \in T_3 \text{ and} \qquad h_i \leq \sqrt{\varepsilon} \quad \text{for } r_i \in T_4.$$

It turns out that packing the items in T_2 involves the most difficulties. We will therefore consider different cases for packing items in T_2, according to the total height of these items. For all cases we need to pack $T_1 \cup T_3 \cup T_4$ afterwards, using the following lemma. We denote packing heights with the letter g to distinguish them from the height h_i of item r_i.

Lemma 16. *Let R be a rectangle of size $(1, g)$ and let T be a set of items that all have area at most ε such that $T_2 = \emptyset$. We can find a packing of a selection $T' \subseteq T$ into R in time $\mathcal{O}(n \log n)$ such that $T' = T$ or*

$$\mathcal{A}(T') \geq \frac{2}{3}(g - \sqrt{\varepsilon}) - \varepsilon.$$

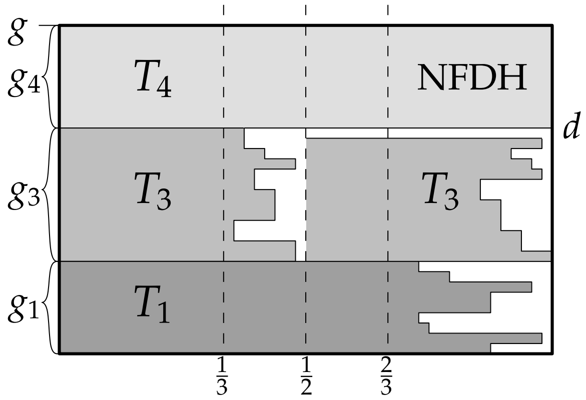


Figure 18: Packing the sets T_1, T_3 and T_4 into a rectangle of width 1 and height g. The difference in height between the stacks of T_3 is denoted by d.

Proof. See Figure 18 for an illustration of the following packing. Stack the items of T_1 left-justified into the lower left corner of R. Stop if there is not sufficient space to accomodate the next item. In this case a total area of at least $\mathcal{A}(T_1') \geq 2/3\,(g - 3/2 \cdot \varepsilon)$ is packed since $w_i > 2/3$ and $h_i \leq 3/2 \cdot \varepsilon$ for items in T_1.

Thus assume all items from T_1 are packed. Denote the height of the stack by g_1. Obviously, $\mathcal{A}(T_1) \geq 2/3 \cdot g_1$.

Create two stacks of items from T_3 next to each other directly above the stack for T_1 by repeatedly assigning each item to the lower stack. Stop if an item does not fit into the rectangle. In this case both stacks have height at least $g - g_1 - 3\varepsilon$ as otherwise a further item could be packed. Therefore $\mathcal{A}(T_1 \cup T_3') \geq 2/3(g - 3\varepsilon) \geq 2/3 \cdot (g - \sqrt{\varepsilon})$ since $3\varepsilon \leq \sqrt{\varepsilon}$.

Otherwise denote the height of the higher stack by g_3 and the height difference by d. The total area of T_3 is at least $\mathcal{A}(T_3) \geq 2/3(g_3 - d) + 1/3 \cdot d \geq 2/3 \cdot g_3 - 1/3 \cdot d \geq 2/3 \cdot g_3 - \varepsilon$ since $w_i \geq 1/3$ and $h_i \leq 3\varepsilon$ for $r_i \in T_3$.

Finally, let $g_4 := g - g_1 - g_3$ and add the items of T_4 by NFDH into the remaining rectangle of size $(1, g_4)$. Lemma 1 yields that either all items are packed, i.e., $T' = T$, or items $T_4' \subseteq T_4$ of total area at least $\mathcal{A}(T_4') \geq 2/3(g_4 - \sqrt{\varepsilon})$ are packed. Thus the total area of the packed items T' is $\mathcal{A}(T') \geq 2/3 \cdot g_1 + 2/3 \cdot g_3 - \varepsilon + 2/3(g_4 - \sqrt{\varepsilon}) \geq 2/3\,(g - \sqrt{\varepsilon}) - \varepsilon$.

The running time is dominated by the application of NFDH. □

If $T_4 = \emptyset$ then the last packing step is obsolete and the analysis above yields the following corollary.

Corollary 6. *Let R be a rectangle of size $(1, g)$ and let T be a set of items that all have area at most ε such that $T_2 \cup T_4 = \emptyset$. We can find a packing of a selection $T' \subseteq T$ into R in time $\mathcal{O}(n)$ such that $T' = T$ or*

$$\mathcal{A}(T') \geq \frac{2}{3}g - 2\varepsilon.$$

The above packings are very efficient if there are no items of width within 1/2 and 2/3 as they essentially yield a width guarantee of 2/3 for the whole height, except for some wasted height that is suitably bounded. In order to pack items of T_2, we have to consider both possible orientations to achieve a total area of more than 1/2 in a packing. We are now ready to prove Lemma 15 that we already presented in the previous section. It shows how sets of items including items of width within 1/2 and 2/3 are processed.

Proof of Lemma 15. Let g_2 be the total height of items in T_2. We present three methods for packing T depending on g_2. For each method we give a lower bound on the total area of items that are packed. Afterwards we show that there cannot be any item that remains unpacked. Throughout the proof, we assume that we do not run out of items while packing the items in T. This will eventually lead to a contradiction in all three cases. Let A be the area of the packed items for which we want to derive lower bounds.

CASE 1. $g_2 \leq 1/3$
Stack the items of T_2 left-justified into the lower left corner of the bin. Use Lemma 16 to pack $T_1 \cup T_3 \cup T_4$ into the rectangle $(1, 1-g_2)$ above the stack—see Figure 19. We get an overall packed area of

$$\begin{aligned} A \geq \frac{g_2}{2} + \frac{2}{3}\left(1 - g_2 - \sqrt{\varepsilon}\right) - \varepsilon &= \frac{2}{3} - \frac{g_2}{6} - \varepsilon - \frac{2}{3}\sqrt{\varepsilon} \\ &\geq \frac{11}{18} - \varepsilon - \frac{2}{3}\sqrt{\varepsilon} \qquad \text{as } g_2 \leq \frac{1}{3}. \end{aligned}$$

CASE 2. $g_2 \in (1/3, 2/3]$
Stack the items of T_2 left-justified into the lower left corner of the bin. Let $B = (1/3, g_2)$ be the free space to the right of the stack. We are going to pack items from $X = \{r_i \in T_3 \cup T_4 \mid w_i \leq g_2\}$ into B. Take an item from X and add it to an initially empty set X' as long as X is nonempty and $\mathcal{A}(X') \leq g_2/6 - \varepsilon$. Rotate the items in X' and use Steinberg's algorithm to pack them into B. This is possible by Corollary 5 since the area of B is $g_2/3$, $\mathcal{A}(X') \leq g_2/6$, and $h_i \leq g_2$ and $w_i \leq \sqrt{\varepsilon} \leq 1/6$ for $r_i \in X'$ (w_i and h_i are the rotated lengths of r_i). Use Lemma 16 to pack $(T_1 \cup T_3 \cup T_4) \setminus X'$ into the rectangle $(1, 1-g_2)$ above the stack—see Figure 19. We distinguish two cases. If $\mathcal{A}(X') \geq g_2/6 - \varepsilon$, then

$$A \geq \overbrace{\frac{g_2}{2}}^{T_2} + \overbrace{\frac{g_2}{6} - \varepsilon}^{X'} + \overbrace{\frac{2}{3}\left(1 - g_2 - \sqrt{\varepsilon}\right) - \varepsilon}^{(T_1 \cup T_3 \cup T_4)\setminus X'} = \frac{2}{3} - 2\varepsilon - \frac{2}{3}\sqrt{\varepsilon}.$$

Otherwise, $\mathcal{A}(X') < g_2/6 - \varepsilon$ and since no further item was added to X' we have $X' = X$. As $g_2 > 1/3$ we have $T_4 \subseteq X$ and we can apply Corollary 6 to get a total area of

$$\begin{aligned} A \geq \frac{g_2}{2} + \frac{2}{3}\left(1 - g_2\right) - 2\varepsilon &= \frac{2}{3} - \frac{g_2}{6} - 2\varepsilon \\ &\geq \frac{5}{9} - 2\varepsilon \qquad \text{as } g_2 \leq \frac{2}{3}. \end{aligned}$$

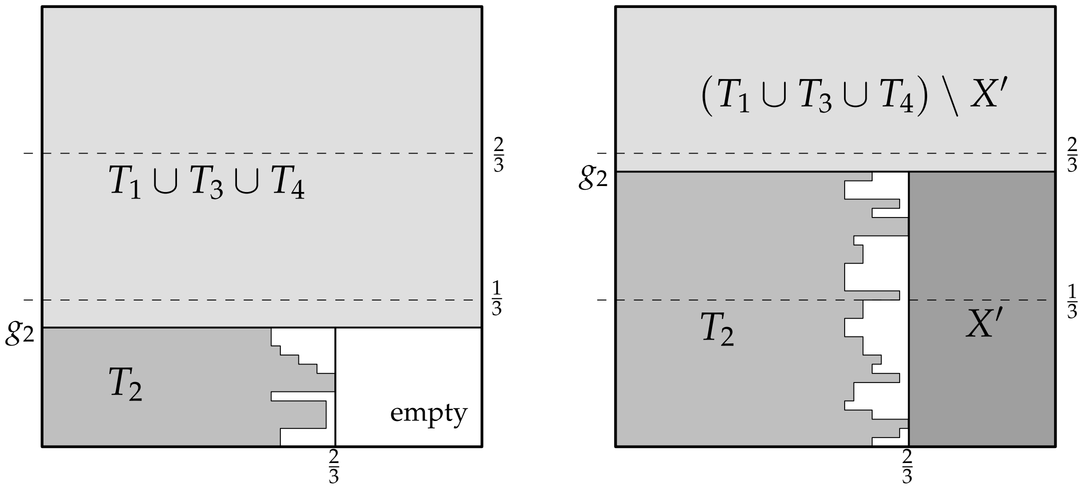


Figure 19: Packing in Case 1 ($g_2 \leq 1/3$) and Case 2 ($1/3 < g_2 \leq 2/3$)

CASE 3. $g_2 \in (2/3, 1 + 4\varepsilon]$
See Figure 20 for an illustration of the following packing and the notations. Order the items of T_2 by non-increasing order of width. Stack the items left-justified into the lower left corner of the bin while the current height g is less or equal to the width of the last item that was packed. In other words, the top right corner of the last item of this stack is above the line from $(1/2, 1/2)$ to $(2/3, 2/3)$, whereas the top right corners of all other items in the stack are below this line. Denote the height of the stack by g and the set of items that is packed into this stack by X_1. Let $r' = (w', h')$ be the last item on the stack. Clearly, $w_i \leq g$ for all items $r_i \in T_2 \setminus X_1$.

Consider the free space $B = (1/3, g)$ to the right of the stack. Rotate the items in $T_2 \setminus X_1$ and stack them horizontally, bottom-aligned into B. Stop if an item does not fit. We denote the items that are packed into B by X_2. Rotate the remaining items $T_2 \setminus (X_1 \cup X_2)$ back into their original orientation and stack them on top of the first stack X_1. Let this set of items be X_3 and let the total height of the stack $X_1 \cup X_3$ be $\widehat{g}$. Use Lemma 16 to pack $T_1 \cup T_3 \cup T_4$ into the rectangle $(1, 1 - \widehat{g})$ above the stack $X_1 \cup X_3$.

Since $w_i \geq g - h'$ for $r_i \in X_1 \setminus \{r'\}$ we have $\mathcal{A}(X_1) \geq (g - h')^2 + h'/2$. Again we distinguish two cases for the analysis. If $X_3 = \emptyset$ (or equivalently $\widehat{g} = g$), then $\mathcal{A}(X_2) \geq (g_2 - g)/2$ and therefore

$$\begin{aligned} A &\geq \overbrace{(g-h')^2 + \frac{h'}{2}}^{X_1} + \overbrace{\frac{g_2 - g}{2}}^{X_2} + \overbrace{\frac{2}{3}\left(1 - g - \sqrt{\varepsilon}\right) - \varepsilon}^{T_1 \cup T_3 \cup T_4} \\ &> (g-h')^2 + \frac{h'}{2} + \frac{1}{3} - \frac{g}{2} + \frac{2}{3}\left(1 - g - \sqrt{\varepsilon}\right) - \varepsilon =: A_1 \qquad \text{as } g_2 > \frac{2}{3}. \end{aligned}$$

To find a lower bound for the total packed area we consider the partial derivative of A_1 with respect to h', which is $\frac{\partial}{\partial h'}(A_1) = 2h' - 2g + 1/2$. Since $2h' - 2g + 1/2 < 0$ for $h' \leq 2\varepsilon$ and $g \geq 1/2$, the total packed area is minimized for the maximal value $h' = 2\varepsilon$ for any g in the domain. After inserting this value for h' we get $A_1 = (g - 2\varepsilon)^2 + \varepsilon +$

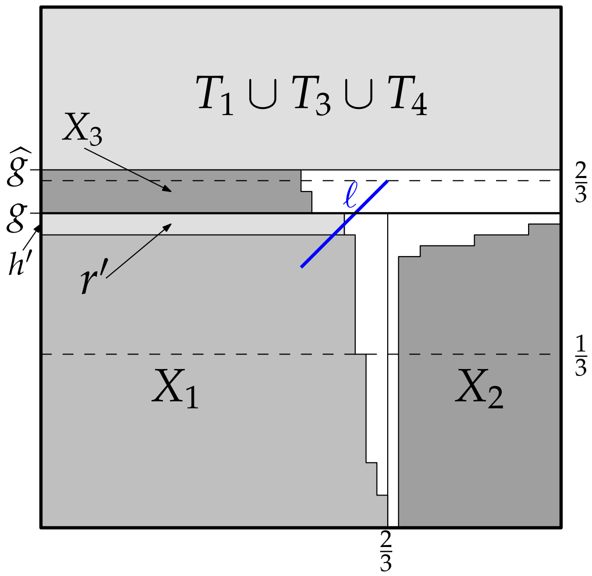


Figure 20: Packing in Case 3 ($2/3 < g_2 \leq 1+4\varepsilon$). Item r' of height h' is depicted larger than $\varepsilon \leq 1/68$ for the sake of visibility. The diagonal line ℓ shows the threshold at which the stack X_1 is discontinued

$1/3 - g/2 + 2/3 \cdot (1 - g - \sqrt{\varepsilon}) - \varepsilon$ and $\frac{\partial}{\partial g}(A_1) = 2g - 7/6 - 4\varepsilon$. Thus the minimum is acquired for $g = 7/12 + 2\varepsilon$. We get

$$\begin{aligned} A_1 &\geq \left(\frac{7}{12}\right)^2 + \varepsilon + \frac{1}{3} - \frac{7}{24} - \varepsilon + \frac{2}{3}\left(\frac{5}{12} - 2\varepsilon - \sqrt{\varepsilon}\right) - \varepsilon \\ &= \frac{95}{144} - \frac{7}{3}\varepsilon - \frac{2}{3}\sqrt{\varepsilon}. \end{aligned}$$

Otherwise $X_3 \neq \emptyset$ (or equivalently $\widehat{g} > g$) and thus $\mathcal{A}(X_2) \geq 1/2 \cdot (1/3 - 2\varepsilon)$ as the stack X_2 leaves at most a width of 2ε of B unpacked. Furthermore, $\widehat{g} \leq 2/3 + 6\varepsilon$ since $g_2 \leq 1 + 4\varepsilon$ and a width of at least $1/3 - 2\varepsilon$ is packed into B. Since $\mathcal{A}(X_3) \geq (\widehat{g} - g)/2$ and $\widehat{g} \leq 2/3 + 6\varepsilon$ we get

$$\begin{aligned} A &\geq \overbrace{(g-h')^2 + \frac{h'}{2}}^{X_1} + \overbrace{\frac{1}{2}\left(\frac{1}{3} - 2\varepsilon\right)}^{X_2} + \overbrace{\frac{\widehat{g}-g}{2}}^{X_3} + \overbrace{\frac{2}{3}\left(1 - \widehat{g} - \sqrt{\varepsilon}\right) - \varepsilon}^{T_1 \cup T_3 \cup T_4} \\ &\geq (g-h')^2 + \frac{h'}{2} + \frac{1}{2}\left(\frac{1}{3} - 2\varepsilon\right) - \frac{1}{9} - \varepsilon - \frac{g}{2} + \frac{2}{3}\left(1 - \sqrt{\varepsilon}\right) - \varepsilon := A_2. \end{aligned}$$

With an analysis similar to before we see that A_2 is minimal for $h' = 2\varepsilon$ (as $\frac{\partial}{\partial h'}(A_2) = 2h' - 2g + 1/2 < 0$) and $g = 1/2$ (as $\frac{\partial}{\partial g}(A_2) = 2g - 1/2 - 4\varepsilon > 0$ for $h = 2\varepsilon$ and $g > 1/2$). We get

$$\begin{aligned} A_2 &\geq \left(\frac{1}{2} - 2\varepsilon\right)^2 + \varepsilon + \frac{1}{6} - \varepsilon - \frac{1}{9} - \varepsilon - \frac{1}{4} + \frac{2}{3}\left(1 - \sqrt{\varepsilon}\right) - \varepsilon \\ &\geq \frac{13}{18} + 4\varepsilon^2 - 4\varepsilon - \frac{2}{3}\sqrt{\varepsilon}. \end{aligned}$$

If $g_2 > 1 + 4\varepsilon$ then $A(T_2) \geq 1/2 \cdot g_2 > 1/2 + 2\varepsilon$, which is a contradiction to the assumption of the lemma. Therefore the three cases cover all possibilities.

It is easy to verify that for $0 < \varepsilon \leq 1/68$ the following inequalities hold.

$$\frac{11}{18} - \varepsilon - \frac{2}{3}\sqrt{\varepsilon} \geq \frac{1}{2} + \varepsilon \qquad \frac{2}{3} - 2\varepsilon - \frac{2}{3}\sqrt{\varepsilon} \geq \frac{1}{2} + \varepsilon$$

$$\frac{5}{9} - 2\varepsilon \geq \frac{1}{2} + \varepsilon \qquad \frac{95}{144} - \frac{7}{3}\varepsilon - \frac{2}{3}\sqrt{\varepsilon} \geq \frac{1}{2} + \varepsilon$$

$$\frac{13}{18} + 4\varepsilon^2 - 4\varepsilon - \frac{2}{3}\sqrt{\varepsilon} \geq \frac{1}{2} + \varepsilon$$

Now let us assume that we do not run out of items while packing a set T with the appropriate method above. Then the packed area is at least $1/2 + \varepsilon$ as the inequalities above show. The contradiction follows from the precondition that removing an arbitrary item from T yields a remaining total area of less than $1/2 + \varepsilon$. Thus all items are packed.

The running time is dominated by the application of Steinberg's algorithm[43]. □

4.1.4 the approximation ratio

Theorem 7. *Our algorithm is an approximation algorithm for two-dimensional bin packing with rotations with an absolute worst case ratio of* 2.

Proof. Recall that we denote the number of bins used for an optimal packing of the large items by ℓ. Obviously $\ell \leq \mathrm{OPT}(I)$. Let s be the number of bins used for packing only small items. If $s \leq \ell$, then the total number of bins is $\ell + s \leq 2\ell \leq 2\,\mathrm{OPT}(I)$. If $s > \ell$, then at least one bin is used for small items and thus all bins for large items contain items with a total area of at least $1/2 - \varepsilon$. According to the partition of the remaining small items, all but the last bin for the small items contain items with a total area of at least $1/2 + \varepsilon$. Let $f > 0$ be the area of the items contained in the last bin. Then

$$\mathrm{OPT}(I) \geq \mathcal{A}(I) \geq \ell \cdot \left(\frac{1}{2} - \varepsilon\right) + (s-1) \cdot \left(\frac{1}{2} + \varepsilon\right) + f > (s + \ell - 1) \cdot \frac{1}{2}.$$

Thus $s + \ell < 2\,\mathrm{OPT}(I) + 1$ and we get $s + \ell \leq 2\,\mathrm{OPT}(I)$ which proves the theorem. □

5 BIN PACKING WITHOUT ROTATIONS

In this chapter we present our results on two-dimensional bin packing without rotations. First, we give a 2-approximation algorithm which is optimal unless $\mathcal{P} = \mathcal{NP}$. Note that this result was independently achieved by Jansen, Prädel & Schwarz[26]. Afterwards, in Section 5.2.1 we affirmatively answer a conjecture by Zhang[47] on the absolute approximation ratio of the Hybrid First Fit (HFF) algorithm by showing that this ratio is 3.

5.1 A 2-APPROXIMATION ALGORITHM FOR BIN PACKING WITHOUT ROTATIONS

Similarly to the approach of our algorithm for bin packing with rotations in Section 4.1, we use the bin packing algorithm by Bansal, Caprara & Sviridenko[6] (here in the variant where rotations are forbidden) to solve instances I with large optimal value. As the asymptotic approximation ratio of their algorithm is arbitrarily close to $1.525\ldots$, there exists a constant k such that for any instance I with optimal value larger than k, their algorithm gives a solution of value at most $2\,\mathrm{OPT}(I)$. We show how to approximate the problem within an absolute factor of 2, provided that the optimal value of the given instance is less than k. Combined with the algorithm by Bansal et al., this proves the existence of an algorithm with an absolute approximation ratio of 2.

Our approach for packing instances I with $\mathrm{OPT}(I) < k$ consists of two parts. First, we give an algorithm that is able to pack instances I with $\mathrm{OPT}(I) = 1$ in two bins in Section 5.1.1 and second, we show how to approximate instances with $1 < \mathrm{OPT}(I) < k$ within a factor of 2 in Section 5.1.2. This at first glance surprising distinction is due to the inherent difficulty of packing wide and high items together into a single bin. In the case $\mathrm{OPT}(I) = 1$ we cannot ensure a separation of the wide and high items into easily feasible sets whereas for $\mathrm{OPT}(I) > 1$ this is possible in many cases.

The approach to solve instances with optimal value greater than some constant k with an asymptotic algorithm is similar to the 2-approximation for two-dimensional bin packing with rotations that we present in Section 4.1, but the methods we use here to handle the instances with smaller optimal value are much more involved. The reason for this is that we cannot use rotations to avoid the necessity to combine wide and high items in a bin. Our approach for solving instances I with $1 < \mathrm{OPT}(I) < k$ is comparable to the main algorithm in Section 4.1 as it is also based on an enumeration of the large items. However, a new ingredient here is a separation of the wide and high items after this enumeration. Another crucial novelty in our algorithm is the use of the $\mathcal{PTAS}$ from [5] to ensure a good area guarantee for at least one bin. In total we show the following theorem.

Theorem 8. *There exists a polynomial-time approximation algorithm for two-dimensional bin packing with absolute approximation ratio* 2*.*

5.1.1 PACKING INSTANCES THAT FIT INTO ONE BIN

Throughout this section we assume that the given instance I can be packed into a single bin, i.e., $\mathrm{OPT}(I) = 1$. At first glance it seems surprising that packing such an instance into two bins is difficult. However, we need to carefully analyze different cases to be able to give a polynomial-time algorithm that solves this problem. In this section we also use the methods that we already introduced in Section 3.1 for strip packing.

Let $\varepsilon := 1/52$. In a first step we consider instances I that satisfy the requirements of Lemma 5 for $x = 0$, i.e., we have $h(W_{1-\delta}) \leq f(\delta) = (\delta - \varepsilon)/(1 + 2\delta)$ for some $\delta \in (\varepsilon, 1/2]$. As we are interested in packing into two bins and not into a strip in a certain direction, we can apply Lemma 5 to the high items instead of the wide items as well. We get the following lemma from Inequality (3.2) for

$$\xi := 0.075 < \frac{1}{4}(1 - \ln 2) - \frac{1}{2}\varepsilon \ln 2 = \xi(0)$$

(here we need $\varepsilon = 1/52$).

Lemma 17. *For any input which cannot be packed in two bins by the methods of Lemma 5, we have*

$$\mathcal{A}(W \cup H) \geq 2\xi + \frac{w(H) + h(W)}{2}.$$

It is crucial for our work that we get this additional area guarantee of $2\xi = 0.15$ on top of the trivial guarantee of $w(H)/2 + h(W)/2$ here. We use this area guarantee to give different methods to pack the input, depending on the total height of the wide items. To do this, we assume that we have $h(W) \geq w(H)$ by otherwise rotating the whole instance and apply different methods for $w(H) > 1/2$ and $w(H) \leq 1/2$. In all cases we are able to pack the input into at most two bins. Before we show how to solve both cases above we need the following lemma that allows us to pack *all* wide items and high items of almost half of their total width, i.e., that shows that we can approximate a packing of Lemma 2 arbitrarily well.

Lemma 18. *For any fixed $\delta > 0$, there exists a polynomial-time algorithm that, given sets W and H of wide and high items with* $\mathrm{OPT}(W \cup H) = 1$*, returns a packing of $W \cup H'$ into a bin with $H' \subseteq H$ and $w(H') > w(H)/2 - \delta$.*

Proof. By interchanging the roles of the wide and the high items in Lemma 2, we get that a packing of $W \cup H^*$ exists with $H^* \subseteq H$ and $w(H^*) \geq w(H)/2$. Let $H^*_{\geq\delta} = \{r_i \in H^* \mid w_i \geq \delta\}$ and $H^*_{<\delta} = \{r_i \in H^* \mid w_i < \delta\}$. We approximate $H^* = H^*_{\geq\delta} \cup H^*_{<\delta}$ as follows.

First, pack the items of W in a stack by non-increasing order of width and align this stack with the bottom right corner of the bin. Second, guess the set $H^*_{\geq\delta}$. By guessing we mean that we enumerate all subsets of $\{r_i \in H \mid w_i \geq \delta\}$ (which is possible as

$|\{r_i \in H \mid w_i \geq \delta\}| \leq 1/\delta$) and apply the remainder of this algorithm on all these sets. As we eventually consider $H^*_{\geq\delta}$, we assume that we can guess this set. Pack $H^*_{\geq\delta}$ into a stack by non-increasing order of height and align this stack with the top left corner of the bin. Third, we approximate $H^*_{<\delta}$ by greedily inserting items from $H_{<\delta} = \{r_i \in H \mid w_i < \delta\}$ into this stack. To do this, start with $H' = H^*_{\geq\delta}$. Now sort the items of $H_{<\delta}$ by non-increasing order of height and for each item try to insert it into the stack (at the correct position to preserve the order inside the stack). If this is possible, the item is added to H'.

Assume that $H_{<\delta} = \{r_1, \dots, r_m\}$ with $h_1 \leq \cdots \leq h_m$. Let $v_i = w(\{r_j \in H' \mid h_j \geq h_i\})$ and $v_i^* = w(\{r_j \in H^* \mid h_j \geq h_i\})$. Whenever an item r_i is not inserted in the stack we have $v_i > v_i^* - w_i$. To see this, assume that $v_i^* \geq v_i + w_i$. This means that the substack from H^* of items of height at least h_i has width larger than the substack of items of height at least h_i from the stack of H' *plus* the width of r_i. As all further items in H^* and H' correspond to each other, r_i does not cause a conflict. Now it is easy to see by induction that $w(H') > w(H^*) - \delta$ at the end. □

With these preparations, the following lemma is easy to show.

Lemma 19. *Let $\varepsilon > 0$ and let I be an instance with $\mathrm{OPT}(I) = 1$, $h(W) \geq w(H) > 1/2$, and $h(W_{1-\delta}) > f(\delta)$ and $w(H_{1-\delta}) > f(\delta)$ for all $\delta \in (\varepsilon, 1/2]$. There exists a polynomial-time algorithm that returns a packing of I into two bins.*

Proof. See Figure 21 for an illustration of this case. We use Lemma 18 to pack $W \cup H'$ with $H' \subseteq H$ and $w(H') > w(H)/2 - \varepsilon$ in the first bin. Build a stack of the remaining high items $H \setminus H'$ and align it with the left side of the second bin. The width of this stack is $w(H \setminus H') < w(H)/2 + \varepsilon$. Note that $w(H \setminus H') \leq 1/2$, as otherwise $h(W) \geq w(H) \geq 1 - 2\varepsilon$ and $\mathcal{A}(W \cup H) \geq 2\xi + (w(H) + h(W))/2 \geq 2\xi + 1 - 2\varepsilon > 1$ (by Lemma 17) which is a contradiction to $\mathrm{OPT}(I) = 1$. Pack the remaining items T with Steinberg's algorithm in the free rectangle of size (a, b) with $a = 1 - w(H \setminus H')$ and $b = 1$ next to the stack of $H \setminus H'$. This is possible since $w_{\max}(T) \leq 1/2 \leq 1 - w(H \setminus H')$, $h_{\max}(T) \leq 1/2$ and with Lemma 17 we have

$$\begin{aligned}
2\mathcal{A}(T) &\leq 2\Big(1 - 2\xi - \frac{w(H) + h(W)}{2}\Big) \\
&\leq 2 - 4\xi - \frac{w(H)}{2} - \frac{w(H)}{2} - \frac{1}{2} && \text{as } h(W) \geq \frac{1}{2} \\
&< \frac{3}{2} - 4\xi - w(H \setminus H') + \varepsilon - \frac{1}{4} && \text{as } \frac{w(H)}{2} \geq w(H \setminus H') - \varepsilon \\
& && \text{and } \frac{w(H)}{2} > \frac{1}{4} \\
&< 1 - w(H \setminus H') && \text{as } 4\xi > \frac{1}{4} + \varepsilon \\
&= ab - (2w_{\max}(T) - a)_+(2h_{\max}(T) - b)_+ && \text{as } 2h_{\max}(T) - b \leq 0.
\end{aligned}$$

□

In the following we assume that $w(H) \leq 1/2$ as otherwise we could pack the instance into two bins with the algorithms of Lemma 5 or Lemma 19. Furthermore, we still have our initial assumption $h(W) \geq w(H)$. The following lemma shows that certain sets of small items can be packed together with the set of wide items.

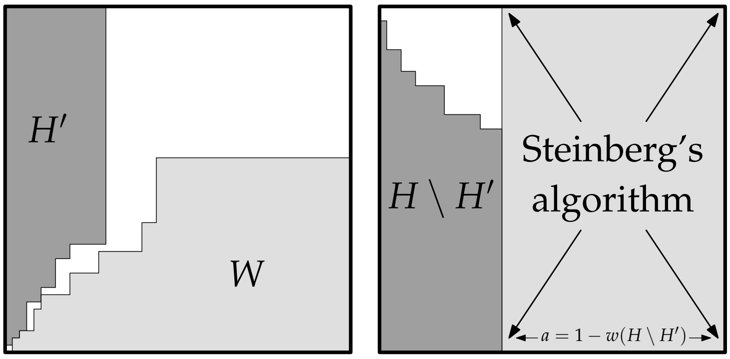


Figure 21: Packing if $h(W) \geq w(H) > 1/2$ and $h(W_{1-\delta}) > f(\delta)$ and $w(H_{1-\delta}) > f(\delta)$ for all $\delta \in (\varepsilon, 1/2]$

Lemma 20. *Any set $T = \{r_1, \ldots, r_m\}$ where $r_i = (w_i, h_i)$ with $w_i \leq 1/2$, $h_i \leq 1 - h(W)$ for $i = 1, \ldots, m$ and total area $\mathcal{A}(T) \leq 1/2 - h(W)/2$ can be packed together with W into a bin.*

Proof. Pack W into a stack of height $h(W)$ and align this stack with the bottom of the bin. Use Steinberg's algorithm to pack T into the free rectangle of size (a, b) with $a = 1$ and $b = 1 - h(W)$ above W. This is possible since $w_{\max}(T) \leq 1/2$, $h_{\max}(T) \leq 1 - h(W)$ and $2\mathcal{A}(T) \leq 1 - h(W) = ab = ab - (2w_{\max}(T) - a)_+(2h_{\max}(T) - b)_+$ as $2w_{\max}(T) - a \leq 0$. □

Obviously, Lemma 20 can also be formulated such that we pack the high items together with a set of small items of total area at most $1/2 - w(H)/2$ (in this case we do not need a condition like $h_i \leq 1 - h(W)$, as $w(H) \leq 1/2$ and thus all remaining items fit into the free rectangle next to the stack of H). This suggests partitioning the small items into sets with these area bounds in order to pack them with the wide and high items. Before we can show how a partition of the small items into sets that satisfy the requirement of Lemma 20 can be generated, we consider two cases where we have to apply a different packing. In all cases, we pack W in a stack of height $h(W)$ in the bottom right corner of the first bin and pack H in a stack of width $w(H)$ in the top left corner of the second bin.

Let ω be the greatest width in the stack of W that is packed above height $1/2$ in this packing (let $\omega = 1/2$ if $h(W) \leq 1/2$) (see Figure 22). We consider the set $\widetilde{H} = \{r_i \mid h_i \in (1 - h(W), 1/2]\}$, i.e., the set of remaining items that do not fit above the stack of the wide items (and thus violate the condition of Lemma 20). Since $\widetilde{H} = \emptyset$ for $h(W) \leq 1/2$, the following case can only occur if $h(W) > 1/2$.

CASE 1. $w(\widetilde{H}) \geq (1 - \omega)/2$.
If there is an item $r_i = (w_i, h_i) \in \widetilde{H}$ with $w_i > (1 - \omega)/2$ then we pack this item in the top left corner of the first bin. It is easy to see that this is possible since $\mathrm{OPT}(I) = 1$. Otherwise greedily pack items from $\widetilde{H}$ into a horizontal stack in the top left corner of the first bin as long as they fit. Since all items in $\widetilde{H}$ have width at most $(1 - \omega)/2$ we can pack a total width of at least $(1 - \omega)/2$ before a conflict occurs.

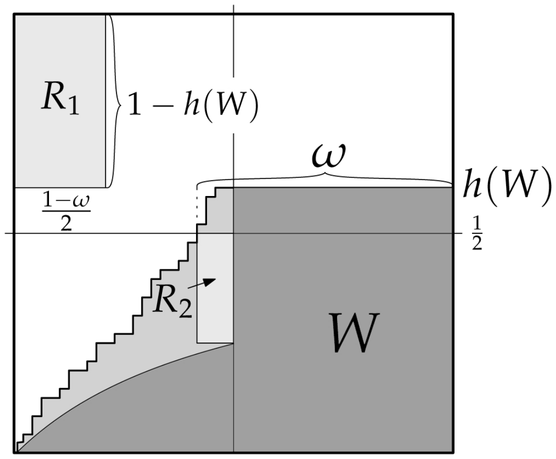


Figure 22: The greatest width in the stack of W that is packed above height $1/2$ is denoted by ω. The additional area guarantees for Case 1 of $\mathcal{A}(R_1) = (1 - h(W))(1 - \omega)/2$ and $\mathcal{A}(R_2) = (\omega - 1/2)/4$ are drawn with lighter shading. The original area guarantee of Lemma 17 is drawn with darker shading.

In both cases we packed a total area of at least $(1 - h(W))(1 - \omega)/2$ from $\widetilde{H}$ into the first bin (see rectangle R_1 in Figure 22). Furthermore, we can use the definition of ω to improve the estimate of Lemma 17. In Lemma 17 all wide items that are packed above height $1/4$ only contribute with their trivial area guarantee of half their height. Now we know that the items that are packed between height $1/4$ and $1/2$ have width at least ω and this gives an additional area of at least $(\omega - 1/2) \cdot 1/4$ (see rectangle R_2 in Figure 22). Thus we packed an overall area of

$$\begin{aligned} A &\geq 2\xi + \frac{w(H) + h(W)}{2} + \frac{(1 - h(W))(1 - \omega)}{2} + \frac{\omega - 1/2}{4} \\ &= \frac{3}{8} + 2\xi + \frac{w(H)}{2} + \omega\left(\frac{h(W)}{2} - \frac{1}{4}\right) \\ &\geq \frac{3}{8} + 2\xi + \frac{w(H)}{2} \qquad \text{as } h(W) \geq 1/2 \text{ and } \omega > 0. \end{aligned}$$

Therefore, the remaining items have total area at most $5/8 - 2\xi - w(H)/2 < 1/2 - w(H)/2$. Thus Lemma 20 allows us to pack these items together with the high items in the second bin.

Let r', r'' be the two largest items in $S \setminus \widetilde{H}$, i.e., among the remaining small items S with $h_i \leq 1 - h(W)$.

CASE 2. $\mathcal{A}(\{r', r''\}) \geq 1/2 - 2\xi - h(W)/2$.
Pack r' in the top left and r'' in the top right corner of the first bin. This is possible as the width of both items is at most $1/2$ and their height is at most $1 - h(W)$. By Lemma 17 and the condition for this case the total area of the packed items is

$$A \geq 2\xi + \frac{w(H) + h(W)}{2} + \frac{1}{2} - 2\xi - \frac{h(W)}{2} = \frac{1}{2} + \frac{w(H)}{2}.$$

Again we can use the method of Lemma 20 to pack the remaining items together with H in the second bin.

CASE 3. Otherwise we have $w(\widetilde{H}) < (1-\omega)/2$ and $\mathcal{A}(\{r', r''\}) < 1/2 - 2\xi - h(W)/2$. This yields that $\mathcal{A}(\widetilde{H}) < (1-\omega)/4$ and $\mathcal{A}(\{r''\}) < 1/4 - \xi - h(W)/4$ (where we assume that $\mathcal{A}(\{r'\}) \geq \mathcal{A}(\{r''\})$. Use the following greedy algorithm to partition the remaining items into two sets that will be packed together with W and H using Lemma 20.

1. Create sets S_1 and S_2 with capacities $c_1 = 1/2 - h(W)/2$ and $c_2 = 1/2 - w(H)/2$, respectively,
2. add r' to S_1 and add all items of $\widetilde{H}$ to S_2,
3. take the remaining items by non-increasing order of size and greedily add them to the set of greater remaining free capacity, i.e., to a set with maximal $c_i - \mathcal{A}(S_i)$.

In the following we show that $\mathcal{A}(S_1) \leq c_1$ and $\mathcal{A}(S_2) \leq c_2$. First note that this holds after Step 2 since

$$\begin{aligned} \mathcal{A}(\{r'\}) &< \frac{1}{2} - 2\xi - \frac{h(W)}{2} < \frac{1}{2} - \frac{h(W)}{2} = c_1 \quad \text{and} \\ \mathcal{A}(\widetilde{H}) &< \frac{1-\omega}{4} < \frac{1-\omega}{2} \leq \frac{1}{2} - \frac{w(H)}{2} = c_2 \qquad \text{as } \omega \geq 1/2 \geq w(H). \end{aligned}$$

Assume that after Step 3 one of the sets has total area greater than its corresponding capacity. Then there is an item r^* that has been added in Step 3 and that violates the capacity for the first time. Since $h(W) \geq f(1/2) = 1/4 - \varepsilon/2$ and $\varepsilon < 1/4$ we have $\mathcal{A}(\{r^*\}) \leq \mathcal{A}(\{r''\}) \leq 1/4 - \xi - h(W)/4 \leq 3/16 - \xi + \varepsilon/8 < 0.15$. Assume w.l.o.g. that r^* was added to S_1. Then we have $\mathcal{A}(S_1) > c_1$ and $\mathcal{A}(S_2) \geq c_2 - \mathcal{A}(r^*)$ as otherwise r^* would have been added to S_2. Thus we have

$$\mathcal{A}(S_1) + \mathcal{A}(S_2) > c_1 + c_2 - \mathcal{A}(\{r^*\}) \geq c_1 + c_2 - 0.15.$$

With Lemma 17 we get the contradiction

$$\begin{aligned} \mathcal{A}(S_1) + \mathcal{A}(S_2) \leq 1 - \mathcal{A}(W \cup H) &\leq 1 - 2\xi - \frac{w(H) + h(W)}{2} \\ &= c_1 + c_2 - 2\xi \\ &= c_1 + c_2 - 0.15. \end{aligned}$$

Thus both sets do not violate their capacity constraints and we can use the methods of Lemma 20 to pack S_1 together with W in the first bin and S_2 together with H in the second bin. Thus we showed the following lemma.

Lemma 21. *Let $\varepsilon > 0$ and let I be an instance with* $\mathrm{OPT}(I) = 1$*, $w(H) \leq 1/2$, and $h(W_{1-\delta}) > f(\delta)$ and $w(H_{1-\delta}) > f(\delta)$ for all $\delta \in (\varepsilon, 1/2]$. There exists a polynomial-time algorithm that returns a packing of I into two bins.*

This concludes our algorithm for instances I with $\mathrm{OPT}(I) = 1$ as the Lemmas 5, 19 and 21 cover all the cases.

5.1.2 PACKING INSTANCES THAT FIT INTO A CONSTANT NUMBER OF BINS

In the following we give a brief description of our algorithm that packs the instances I with $2 \leq \mathrm{OPT}(I) < k$ into $2\,\mathrm{OPT}(I)$ bins. Let $\varepsilon := 1/(20k^3 + 2)$.

Let $L = \{r_i \mid w_i h_i > \varepsilon\}$ be the set of *large* items and let $T = \{r_i \mid w_i h_i \leq \varepsilon\}$ be the set of *tiny* items. As defined in Section 2 we refer to items as wide (W), high (H), small (S) and big, according to their side lengths. Note that the terms *large* and *tiny* refer to the area of the items whereas *big*, *wide*, *high* and *small* refer to their widths and heights. Also note that, e.g., an item can be tiny and high, or wide and big at the same time.

We guess $\ell = \mathrm{OPT}(I) < k$ and open 2ℓ bins that we denote by $B_1, \ldots, B_\ell$ and $C_1, \ldots, C_\ell$. By *guessing* we mean that we iterate over all possible values for ℓ and apply the remainder of this algorithm on every value. As there are only a constant number of values, this is possible in polynomial time. We assume that we know the correct value of ℓ as we eventually consider this value in an iteration. For the ease of presentation, we also denote the sets of items that are associated with the bins by $B_1, \ldots, B_\ell$ and $C_1, \ldots, C_\ell$. We will ensure that the set of items that is associated with a bin is feasible and a packing is known or can be computed in polynomial time. To do this we use Corollary 1.

Let I_i^* be the set of items in the i-th bin in an optimal solution. We assume w.l.o.g. that $\mathcal{A}(I_i^*) \geq \mathcal{A}(I_j^*)$ for $i < j$. Then we have

$$\mathcal{A}(I) = \mathcal{A}(I_1^*) + \cdots + \mathcal{A}(I_\ell^*) \leq \ell \cdot \mathcal{A}(I_1^*). \tag{5.1}$$

In a first step, we guess the assignment of the large items to bins. Using this assignment and the $\mathcal{PTAS}$ from [5] we pack a total area of at least $\mathcal{A}(I_1^*) - \varepsilon$ into B_1 and keep C_1 empty. This step has the purpose of providing a good area bound for the first bin and leaving a free bin for later use. We ensure that the large items that are assigned to B_1 are actually packed. For all other bins we reserve B_i for the wide and small items (except the big items) and C_i for the high and big items for $i = 2, \ldots, \ell$. This separation enables us to use Steinberg's algorithm (Corollary 1) to pack up to half of the bins' area. In detail, the first part of the algorithm works as follows.

1. Guess $L_i = I_i^* \cap L$ for $i = 1, \ldots, \ell$.
2. Apply the $\mathcal{PTAS}$ from [5] on $L_1 \cup T$ with $p_i = \mathcal{A}(r_i) \cdot (1/\varepsilon + 1)$ for $r_i \in L_1$, $p_i = \mathcal{A}(r_i)$ for $r_i \in T$ and an accuracy of $\varepsilon^2/(1+\varepsilon)$. Assign the output to bin B_1 and keep an empty bin C_1.
3. For $i = 2, \ldots, \ell$, assign the wide and small items of L_i to B_i (omitting big items) and assign the high and big items of L_i to C_i. That is, $B_i = L_i \setminus H$ and $C_i = L_i \cap H$.
4. For $i = 2, \ldots, \ell$, greedily add tiny wide items from $T \cap W$ by non-increasing order of width to B_i as long as $\mathcal{A}(B_i) \leq 1/2$ and greedily add tiny high items from $T \cap H$ by non-increasing order of height to C_i as long as $w(C_i) \leq 1$.

Corollary 1 shows that using Steinberg's algorithm the bins $B_2, \ldots, B_\ell$ can be packed as there are no wide items and the total area is at most $1/2$. The bins $C_2, \ldots, C_\ell$ can be packed with a simple stack as they contain only high items of total width at most 1.

Observe that in Step 4 we only add to a new bin B_i if the previous bins contain items of total area at least $1/2 - \varepsilon$ and we only add to a new bin C_i if the previous bins contain items of total width at least $1 - 2\varepsilon$ (as the width of the tiny high items is at most 2ε) and thus of total area at least $1/2 \cdot (1 - 2\varepsilon) = 1/2 - \varepsilon$. After the application of this first part of the algorithm, some tiny items $T' \subseteq T$ might remain unpacked. Note that if $\mathcal{A}(B_\ell) < 1/2 - \varepsilon$, then there are no wide items in T' and if $\mathcal{A}(C_\ell) < 1/2 - \varepsilon$ then there are no high items in T' (as these items would have been packed in Step 4). We distinguish different cases to continue the packing according to the filling of the last bins B_ℓ and C_ℓ.

In the following we show that we actually ensure that the large items that are assigned to B_1 are packed into this bin. First note that Theorem 2 can be applied for $r = 1/\varepsilon + 1$ as $p_i/\mathcal{A}(r_i) \in \{1, 1/\varepsilon + 1\}$ for all items in $L_1 \cup T$. Now it is easy to see that L_1 is packed since $p_i > 1 + \varepsilon$ for $r_i \in L_1$, whereas $p(\widetilde{T}) = \mathcal{A}(\widetilde{T}) \leq 1$ for any feasible set $\widetilde{T} \subseteq T$. Thus $L_1 = I_1^* \cap L = B_1 \cap L$. Furthermore, for the set of packed tiny items $B_1 \cap T$ we have

$$\mathcal{A}(B_1 \cap T) \geq \mathcal{A}(I_1^* \cap T) - \varepsilon$$

since $(1/\varepsilon + 1)\mathcal{A}(B_1 \cap L) + \mathcal{A}(B_1 \cap T) = p(B_1)$ and

$$\begin{aligned} p(B_1) &\geq \Big(1 - \frac{\varepsilon^2}{1+\varepsilon}\Big)\mathrm{OPT}(L_1 \cup T) && \text{by Theorem 2} \\ &= \Big(1 - \frac{\varepsilon^2}{1+\varepsilon}\Big)\Big[\Big(\frac{1}{\varepsilon} + 1\Big)\mathcal{A}(I_1^* \cap L) + \mathcal{A}(I_1^* \cap T)\Big] \\ &\geq \Big(\frac{1}{\varepsilon} + 1\Big)\mathcal{A}(I_1^* \cap L) + \mathcal{A}(I_1^* \cap T) - \frac{\varepsilon^2}{1+\varepsilon}\Big(\frac{1}{\varepsilon} + 1\Big) \\ & && \text{as } \Big(\frac{1}{\varepsilon} + 1\Big)\mathcal{A}(I_1^* \cap L) + \mathcal{A}(I_1^* \cap T) \leq \frac{1}{\varepsilon} + 1 \\ &= \Big(\frac{1}{\varepsilon} + 1\Big)\mathcal{A}(B_1 \cap L) + \mathcal{A}(I_1^* \cap T) - \varepsilon. \end{aligned}$$

Thus we have

$$\mathcal{A}(B_1) \geq \mathcal{A}(I_1^*) - \varepsilon. \tag{5.2}$$

Now we are ready to start with the case analysis.

CASE 1. $\mathcal{A}(B_\ell) < 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) < 1/2 - \varepsilon$.
In this case T' does not contain any wide or high items as these items would have been packed to B_ℓ or C_ℓ. Greedily add items from T' into all bins except B_1 as long as the bins contain items of total area at most $1/2$. After adding the items from T', either

all items are assigned to a bin (and can thus be packed) or each bin contains items of total area at least $1/2 - \varepsilon$ and we packed a total area of at least

$$\begin{aligned}
A &\geq \mathcal{A}(B_1) + (2\ell - 1)\Big(\frac{1}{2} - \varepsilon\Big) \\
&\geq \mathcal{A}(I_1^*) + \ell - \frac{1}{2} - 2\ell\varepsilon && \text{by Inequality (5.2)} \\
&\geq \mathcal{A}(I_1^*) + \ell\mathcal{A}(I_1^*) + \ell(1 - \mathcal{A}(I_1^*)) - \frac{1}{2} - 2\ell\varepsilon \\
&\geq \ell\mathcal{A}(I_1^*) + \frac{1}{2} - 2\ell\varepsilon && \text{as } \ell \geq 1 \text{ and } 1 - \mathcal{A}(I_1^*) \geq 0 \\
&> \ell\mathcal{A}(I_1^*) && \text{as } \varepsilon < \frac{1}{4\ell}.
\end{aligned}$$

Since this contradicts Inequality (5.1), all items are packed.

CASE 2. $\mathcal{A}(B_\ell) \geq 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) \geq 1/2 - \varepsilon$.
In this case T' might contain wide and high items. On the other hand the bin C_1 is still available for packing. We use the area of the items in bin C_ℓ to bound the total area of the packed items and (with a similar calculation as in Case 1) we get a packed area of at least $A \geq \mathcal{A}(B_1) + \mathcal{A}(C_\ell) + (2\ell - 3)(1/2 - \varepsilon) \geq \ell\mathcal{A}(I_1^*) + \mathcal{A}(C_\ell) - 1/2 - (2\ell - 2)\varepsilon$. As $\mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$ (Inequality (5.1)) we get

$$\mathcal{A}(T') \leq \mathcal{A}(I) - A \leq \frac{1}{2} + (2\ell - 2)\varepsilon - \mathcal{A}(C_\ell) \qquad \text{and hence} \tag{5.3}$$

$$\mathcal{A}(T') \leq (2\ell - 1)\varepsilon \qquad \text{as } \mathcal{A}(C_\ell) \geq 1/2 - \varepsilon \tag{5.4}$$

Assume that $\mathcal{A}(C_\ell) < 1/2 + (2\ell - 2)\varepsilon$ as otherwise $T' = \emptyset$ by Inequality (5.3)

We consider the set $\widehat{H} = \{r_i \in C_\ell \mid h_i \leq 3/4\}$. If $w(\widehat{H}) \geq (4\ell - 3)\varepsilon$ then remove $\widehat{H}$ from C_ℓ and pack it in a stack in C_1 instead. As we now have $\mathcal{A}(C_\ell) < 1/2 - (2\ell - 1)\varepsilon$ and $\mathcal{A}(T' \setminus W) \leq \mathcal{A}(T') \leq (2\ell - 1)\varepsilon$ by Inequality (5.4), we can pack $T' \setminus W$ together with C_ℓ. The remaining items $T' \cap W$ have total height at most $2(2\ell - 1)\varepsilon$ and thus fit above $\widehat{H}$ into C_1.

Otherwise, there is no item $r' = (w', h')$ in C_ℓ with $h' \leq 3/4$ and $w' \geq (4\ell - 3)\varepsilon$. Let $\widetilde{H} = \{r_i \in C_\ell \cup T' \mid h_i > 3/4\}$. Observe that we have

$$w(\widetilde{H}) \leq \frac{\mathcal{A}(C_\ell \cup T')}{3/4} \leq \frac{4}{3}\Big(\frac{1}{2} + (2\ell - 2)\varepsilon + (2\ell - 1)\varepsilon\Big) = \frac{2}{3} + \Big(\frac{16}{3}\ell - 4\Big)\varepsilon < 1. \tag{5.5}$$

We take *all* high items from $C_\ell \cup T'$ and order them by non-increasing height. Now pack the items greedily into a stack of width up to 1 and pack this stack into C_ℓ. We have $w(C_\ell) \geq 1 - (4\ell - 3)\varepsilon$ as we bounded the total width of the items from $\widetilde{H}$ in Inequality (5.5) and thus all further items have width at most $(4\ell - 3)\varepsilon$ (as otherwise $\widehat{H} \geq (4\ell - 3)\varepsilon$ and we had solved the problem in the previous step). For the remaining items T' we have $h_{\max}(T') \leq 3/4$ and $\mathcal{A}(T') \leq 1/2 - (2\ell - 2)\varepsilon - \mathcal{A}(C_\ell) \leq (4\ell - 7/2)\varepsilon \leq 1/4$ (by Inequality (5.3) and as $\mathcal{A}(C_\ell) \geq w(C_\ell)/2 \geq 1/2 - (2\ell - 3/2)\varepsilon$ and $\varepsilon < 1/(16\ell)$). Thus T' can be packed into bin C_1 using Steinberg's algorithm.

CASE 3. $\mathcal{A}(B_\ell) < 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) \geq 1/2 - \varepsilon$.
If $w(T' \cap H) \leq 1$ then pack $T' \cap H$ in C_1 and proceed as in Case 1.

The subcase where $w(T' \cap H) > 1$ is the most difficult of all four cases. The challenge that we face is that $w(H)$ can be close to ℓ (which is a natural upper bound) but we can only ensure a packed total width of at least $\ell(1 - 2\varepsilon)$ in the bins $C_1, \dots, C_\ell$. So we have to pack high items into the bins $B_2, \dots, B_\ell$. We distinguish two further subcases.

1. Assume that there exists $j \in \{2, \dots, \ell\}$ with $w(L_j \cap H) > 10\ell\varepsilon$, i.e., the total width of the items that are large and high and associated with the j-th bin in an optimal packing is large enough such that moving these items away gives sufficient space for the still unpacked high items.

   Go back to Step 3 in the first part of the algorithm and omit separating the items from L_j. Instead we assign the items from L_j to bin B_j and keep C_j free at the moment. Note that L_j admits a packing into a bin as L_j corresponds to the large items in a bin of an optimal solution. Since $|L_j| \leq 1/\varepsilon$ we can find such a packing in constant time.

   While greedily adding tiny items in Step 4, we skip B_j for the wide items and we continue packing high items in C_1 after we have filled $C_2, \dots, C_\ell$. As we moved high items of total width at least $10\ell\varepsilon$ to B_j and we can pack high items of total width at least $1 - 2\varepsilon$ into each bin, no high items remains after this step. Finally, greedily add remaining tiny items to bins $B_2, \dots, B_\ell$ except B_j, using the area bound $1/2$.

   Now consider the bins C_1 and C_j. Both contain only tiny items, as we moved the large items from C_j to B_j. We packed the tiny items greedily by height and thus all items in C_j have height greater or equal to any item in C_1. Let h' be greatest height in C_1. Then we have $\mathcal{A}(C_j) \geq h'(1 - 2\varepsilon)$. Furthermore, we know that $w(H) > \ell(1 - 2\varepsilon)$. Thus we have

   $$\mathcal{A}(H) > (\ell - 1)(1/2 - \varepsilon) + h'(1 - 2\varepsilon).$$

   If after the modified Step 4 tiny items remain unpacked, then all bins B_i for $i \in \{2, \dots, \ell\} \setminus \{j\}$ have area $\mathcal{A}(B_i) \geq 1/2 - \varepsilon$. By summing up the area of the high items separately we get a total packed area of at least

   $$\begin{aligned} A &\geq \mathcal{A}(B_1) + \overbrace{(\ell - 2)\Big(\frac{1}{2} - \varepsilon\Big)}^{B_i \text{ for } i \in \{2,\dots,\ell\} \setminus \{j\}} + \mathcal{A}(H) \\ &> \mathcal{A}(B_1) + (\ell - 2)\Big(\frac{1}{2} - \varepsilon\Big) + (\ell - 1)\Big(\frac{1}{2} - \varepsilon\Big) + h'(1 - 2\varepsilon) \\ &\geq \mathcal{A}(I_1^*) + \ell - \frac{3}{2} + h' - (2\ell - 2 + 2h')\varepsilon && \text{by Inequality (5.2)} \\ &\geq \mathcal{A}(I_1^*) + \ell\mathcal{A}(I_1^*) + \ell(1 - \mathcal{A}(I_1^*)) - \frac{3}{2} + h' - 2\ell\varepsilon && \text{as } h' \leq 1 \\ &\geq \ell\mathcal{A}(I_1^*) - \frac{1}{2} + h' - 2\ell\varepsilon \end{aligned}$$

   as $\ell \geq 1$ and $1 - \mathcal{A}(I_1^*) \geq 0$.

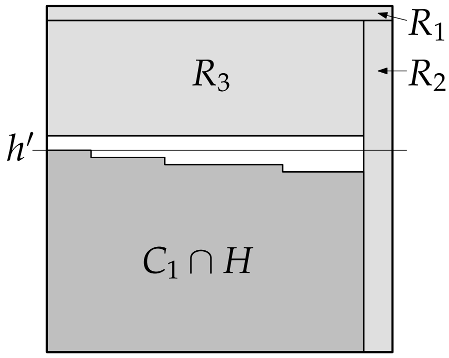


Figure 23: The three rectangles R_1, R_2 and R_3 for packing T' together with $C_1 \cap H$ in C_1.

On the other hand we have $A \leq \mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$ by Inequality (5.1). Thus the total area of the remaining items T' is at most $\mathcal{A}(T') \leq 1/2 + 2\ell\varepsilon - h'$. If $h' \geq 1/2 + 2\ell\varepsilon$ we packed all items.

Otherwise we have $1/2 < h' < 1/2 + 2\ell\varepsilon$ and

$$\mathcal{A}(T') \leq 2\ell\varepsilon. \tag{5.6}$$

We will pack T' in C_1 together with the already packed high items. Observe that

$$w(C_1 \cap H) \leq 1 - 8\ell\varepsilon \tag{5.7}$$

as we move high items of total area of at least $10\ell\varepsilon$ to B_j and all bins $C_2, \ldots, C_\ell$ are filled up to a width of at least $1 - 2\varepsilon$.

We pack the remaining items T' into three rectangles $R_1 = (1, 4\ell\varepsilon)$, $R_2 = (8\ell\varepsilon, 1 - 4\ell\varepsilon)$ and $R_3 = (1 - 8\ell\varepsilon, 1/2 - 6\ell\varepsilon)$ which can be packed in C_1 together with $C_1 \cap H$ as follows—see Figure 23. Pack the stack of $C_1 \cap H$ in the lower left corner and pack R_3 above this stack. As $h' + h(R_3) \leq 1 - h(R_1)$, R_1 fits in the top of C_1. Finally, pack R_2 in the bottom right corner. This is possible as $h(R_2) \leq 1 - h(R_1)$ and $w(C_1 \cap H) \leq 1 - 8\ell\varepsilon = 1 - w(R_3)$ by Inequality (5.7).

Now pack $T' \cap W$ in a stack in R_1 (which is possible since $h(T' \cap W) \leq 2\mathcal{A}(T') \leq 4\ell\varepsilon$ by Inequality (5.6)) and pack all $r_i = (w_i, h_i) \in T'$ with $h_i > 1/2 - 6\ell\varepsilon$ in a vertical stack in R_2 (this fits as the total width of items with $h_i > 1/2 - 6\ell\varepsilon$ is at most $\mathcal{A}(T')/(1/2 - 6\ell\varepsilon) \leq 8\ell\varepsilon$ by Inequality (5.6) and as $6\ell\varepsilon \leq 1/4$). Finally, use Steinberg's algorithm to pack the remaining items in R_3. This is possible since $w_{\max} \leq 1/2$, $h_{\max} \leq 1/2 - 6\ell\varepsilon$ and

$$\begin{aligned} 2\mathcal{A}(T') &\leq 4\ell\varepsilon \\ &\leq \frac{1}{2} - 14\ell\varepsilon + 96\ell^2\varepsilon^2 \\ &= (1 - 8\ell\varepsilon)\Big(\frac{1}{2} - 6\ell\varepsilon\Big) - (1 - 1 + 8\ell\varepsilon)_+\Big(1 - 12\ell\varepsilon - \frac{1}{2} + 6\ell\varepsilon\Big)_+. \end{aligned}$$

This finishes the first case where we assumed that there exists $j \in \{2, \ldots, \ell\}$ with $w(L_j \cap H) > 10\ell\varepsilon$.

2. Now assume that we have $w(L_j \cap H) \leq 10\ell\varepsilon$ for all $j \in \{2, \dots, \ell\}$ and in particular $w_i \leq 10\ell\varepsilon$ for all items $r_i = (w_i, h_i) \in (L_2 \cup \dots \cup L_\ell)$. Thus all high items that are not packed in B_1 are thin, i.e., have width at most $10\ell\varepsilon$. We use this fact by repacking the high items greedily by non-increasing height in the bins $C_1, \dots, C_\ell$. Each bin contains high items of total width at least $1 - 10\ell\varepsilon$ afterwards. Thus high items of total width at most $10\ell^2\varepsilon$ remain unpacked. This is worse than in the previous case but since we repacked all items we can get a nice bound on the height of the unpacked items. Let h' be the smallest height in C_ℓ. Then all items in $C_1, \dots, C_\ell$ have height at least h' and the remaining items T' have height at most h'. If there is an $i \in \{2, \dots, \ell\}$ with $\mathcal{A}(B_i) \leq 1 - h' - 10\ell^2\varepsilon$ then we can add the remaining items T' to B_i using Steinberg's algorithm. To see this note that $h_{\max}(T' \cup B_i) \leq h'$ and $2\mathcal{A}(T' \cup B_i) \leq 2\mathcal{A}(B_i) + 2(10\ell^2\varepsilon)h' \leq 2 - 2h'$ which corresponds to the bound of Theorem 1.

Otherwise for all $i \in \{2, \dots, \ell\}$ we have $\mathcal{A}(B_i) \geq 1 - h' - 10\ell^2\varepsilon$. Then we packed a total area of at least

$$\begin{aligned}
A &\geq \mathcal{A}(B_1) + \mathcal{A}(C_1 \cup \dots \cup C_\ell) + \mathcal{A}(B_2 \cup \dots \cup B_\ell) \\
&\geq \mathcal{A}(B_1) + h'\ell(1 - 10\ell\varepsilon) + (\ell - 1)(1 - h' - 10\ell^2\varepsilon) \\
&\geq \mathcal{A}(I_1^*) + \ell - 1 + h' - (10\ell^3 + 1)\varepsilon \\
&> \mathcal{A}(I_1^*) + \ell\mathcal{A}(I_1^*) + \ell(1 - \mathcal{A}(I_1^*)) - 1 + \frac{1}{2} - (10\ell^3 + 1)\varepsilon \qquad \text{as } h' > \frac{1}{2} \\
&\geq \ell\mathcal{A}(I_1^*) + \frac{1}{2} - (10\ell^3 + 1)\varepsilon \qquad \text{as } \ell \geq 1 \text{ and } 1 - \mathcal{A}(I_1^*) \geq 0.
\end{aligned}$$

And since $1/2 - (10\ell^3 + 1)\varepsilon \geq 0$ and $A \leq \mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$ by Inequality (5.1), no item remains unpacked.

Thus in both subcases we are able to derive a feasible packing.

CASE 4. $\mathcal{A}(B_\ell) \geq 1/2 - \varepsilon$ and $\mathcal{A}(C_\ell) < 1/2 - \varepsilon$.
In this case T' contains no high items. If there are also no wide items remaining in T', apply the methods of Case 1. Otherwise we use the following process to free some space in the bins for wide and small items, i.e., $B_2, \dots, B_\ell$. The idea of the process is to move small items from bins B_i to bins C_i and thereby move the tiny high items $T' \cap H$ further in direction C_ℓ. To do this, let $S_i = L_i \cap S$ be the set of small items in B_i.

Remove the tiny items from $C_2, \dots, C_\ell$. If there exists an item $r \in S_i \cap B_i$ for some $i \in \{2, \dots, \ell\}$ then remove r from B_i and add it to C_i, otherwise stop. Adding r to C_i is possible as C_i is a subset of $L_i = I_i^*$ and thus feasible. Add wide items from $W \cap T'$ to B_i until $\mathcal{A}(B_i) \geq 1/2 - \varepsilon$ again or $W \cap T' = \emptyset$. Finally, add the high items from $H \cap T'$ to $C_2, \dots, C_\ell$ in a greedy manner analogously to Step 4 of the first part of the algorithm but using the area bound $\mathcal{A}(C_i) \leq 1/2$. This ensures that all sets C_i can be packed with Steinberg's algorithm (we use Corollary 1 here as there might be big items in C_i). Repeat this process until $S_i \cap B_i = \emptyset$ for all $i \in \{2, \dots, \ell\}$ or T' contains a high item at the end of an iteration.

There are two ways in which this process can stop. First if we moved all items from S_i to C_i, and second if in the next step a high item would remain in T' after the process.

In the first case we have reached a situation as in Case 2 or Case 3, i.e., the roles of the wide and the high items are interchanged and $\mathcal{A}(B_\ell) \geq 1/2 - \varepsilon$. Thus by rotating all items and the packing derived so far, we can solve this case analogously to Case 2 or Case 3, depending on $\mathcal{A}(C_\ell)$.

In the second case, let r^* be the item that stopped the process, i.e., if r^* is moved from B_i to C_i for some $i \in \{2, \dots, \ell\}$, at least one high item would remain in T'. Then, instead of moving r^* to C_i we move r^* to C_1 and add items from T' to C_1 and C_i as long as $\mathcal{A}(C_1) \leq 1/2$ and $\mathcal{A}(C_i) \leq 1/2$. The resulting sets can be packed with Steinberg's algorithm as no item has width greater than $1/2$. If after this step still items remain unpacked then a calculation similar to Case 1 gives a total packed area of

$$\begin{aligned} A &\geq \mathcal{A}(B_1) + \overbrace{(2\ell - 2)\left(\frac{1}{2} - \varepsilon\right)}^{\text{all bins except } B_1, B_i} + \overbrace{\frac{1}{2} - \varepsilon - \mathcal{A}(r^*)}^{\text{bin } B_i} \\ &\geq \ell\mathcal{A}(I_1^*) + \frac{1}{2} - 2\ell\varepsilon - \mathcal{A}(r^*) > \ell\mathcal{A}(I_1^*) \qquad \text{since } \mathcal{A}(r^*) \leq 1/4 \text{ and } \varepsilon < 1/(8\ell). \end{aligned}$$

As we have a contradiction to $A \leq \mathcal{A}(I) \leq \ell\mathcal{A}(I_1^*)$ by Inequality (5.1), all items are packed.

We showed the following lemma which concludes our presentation of the 2-approximation algorithm for two-dimensional bin packing.

Lemma 22. *There exists a polynomial-time algorithm that, given an instances I with $1 < \mathrm{OPT}(I) < k$, returns a packing in $2\,\mathrm{OPT}(I)$ bins.*

5.2 THE APPROXIMATION RATIO OF HYBRID FIRST FIT IS 3

In this section we prove Zhang's conjecture[47] on the absolute approximation ratio of the Hybrid First Fit (HFF) algorithm by showing that this ratio is 3.

We start our presentation in Section 5.2.1 with a description of the HFF algorithm which was introduced by Chung, Gary & Johnson[12] in 1982 and whose running time is slightly better than the running time of Zhang's previously known 3-approximation algorithm[47]. A lower bound on the approximation ratio is presented in Section 5.2.2. Note that the instance that we give in this section also holds as a lower bound for Zhang's algorithm[47] and for Hybrid First Fit by Width that we describe in Section 5.2.1. Thus it does not suffice to combine all three algorithms in order to derive a better approximation ratio. Finally, we give the proof of the upper bound in Section 5.2.3.

5.2.1 THE HYBRID FIRST FIT ALGORITHM

HFF is based on the one-dimensional First Fit Decreasing (FFD) bin packing algorithm and on the First Fit Decreasing Height (FFDH) strip packing algorithm. The latter algorithm is a layer-based strip packing algorithm similar to NFDH that we introduced in Section 2. It was considered for the first time by Chung, Garey, & Johnson[12] and is given as follows. Sort the items by non-increasing order of height. Pack the items one by one into layers. The height of a layer is defined by its first item, further items are added left-aligned into the lowest layer with sufficient space. If an item does not fit into any layer opened so far this item opens a new layer. HFF now considers the layers of a FFDH packing one after the other and packs each layer into the first bin with sufficient space. Since the layers are ordered by non-increasing height, this corresponds to a one-dimensional FFD packing. See Figure 24 for an illustration of HFF.

A simple variant of HFF, that we denote as Hybrid First Fit by Width packs the strip in the direction of the width. This means that the items are ordered by non-increasing width for the strip packing and FFD is later applied on the width of the resulting layers. This algorithm can also be seen as HFF applied on the instance $J = \{(h_i, w_i) \mid (w_i, h_i) \in I\}$, where each item of I is rotated by 90 degrees. Afterwards the packing is rotated back.

Using a tree data structure from Johnson[32] the running time of HFF is $\mathcal{O}(n \log n)$. This is slightly faster than the running time of Zhang's algorithm[47], which is dominated by the application of Steinbergs' algorithm[43] that, by Theorem 1, runs in time $\mathcal{O}((n \log^2 n) / \log \log n)$.

We denote the layers from FFDH by $L_1, \dots, L_m$ with heights $h_{\max}(L_1) \geq h_{\max}(L_2) \geq \cdots \geq h_{\max}(L_m)$ and total widths $w(L_1), \dots, w(L_m)$. Note that there is no particular order on the widths of the layers. For the sake of simplicity we refer to both the layer and the set of items in the layer by L_i.

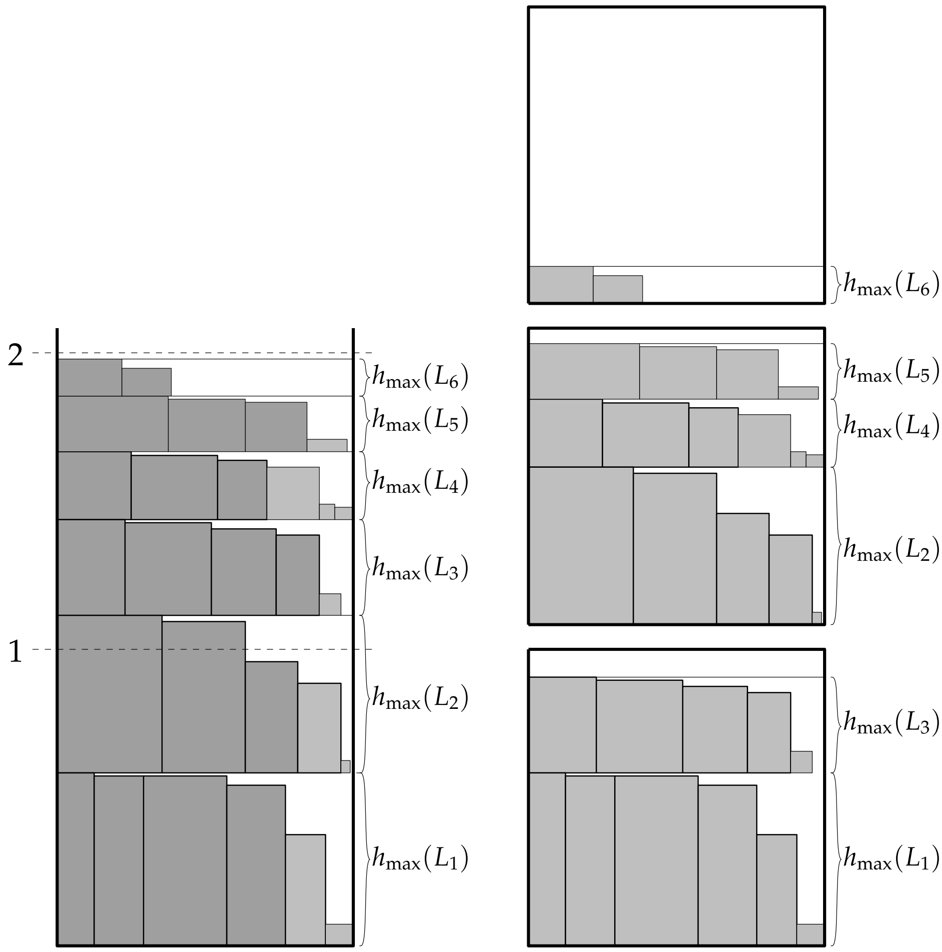


Figure 24: The Hybrid First Fit algorithm. The layers of a strip packing with FFDH are packed into bins with FFD. Items that are placed into a layer after a new layer was opened are shown in light grey

5.2.2 LOWER BOUND

Let $0 < \delta < 1/34$, such that $1/\delta$ is integer, and consider the instance $I = A_1 \cup A_2 \cup B_1 \cup B_2 \cup C_1 \cup C_2$ consisting of the following sets of items

set		items	
A_1	1	item of size	$(\delta, 1-\delta)$
A_2	1	item of size	$(1-\delta, \delta)$
B_1	$\frac{1}{\delta} - 6$	items of size	$(\delta, \frac{1}{2} + \delta)$
B_2	$\frac{1}{\delta} - 6$	items of size	$(\frac{1}{2} + \delta, \delta)$
C_1	3	item of size	$(2\delta, \frac{1}{6} + \frac{1}{3}\delta)$
C_2	3	items of size	$(\frac{1}{6} + \frac{1}{3}\delta, 2\delta)$

In Figure 25 we show that $\text{OPT}(I) = 1$ and we give the FFDH packing of I. We assume that the item in A_2 comes before the items in B_2 in the non-increasing order

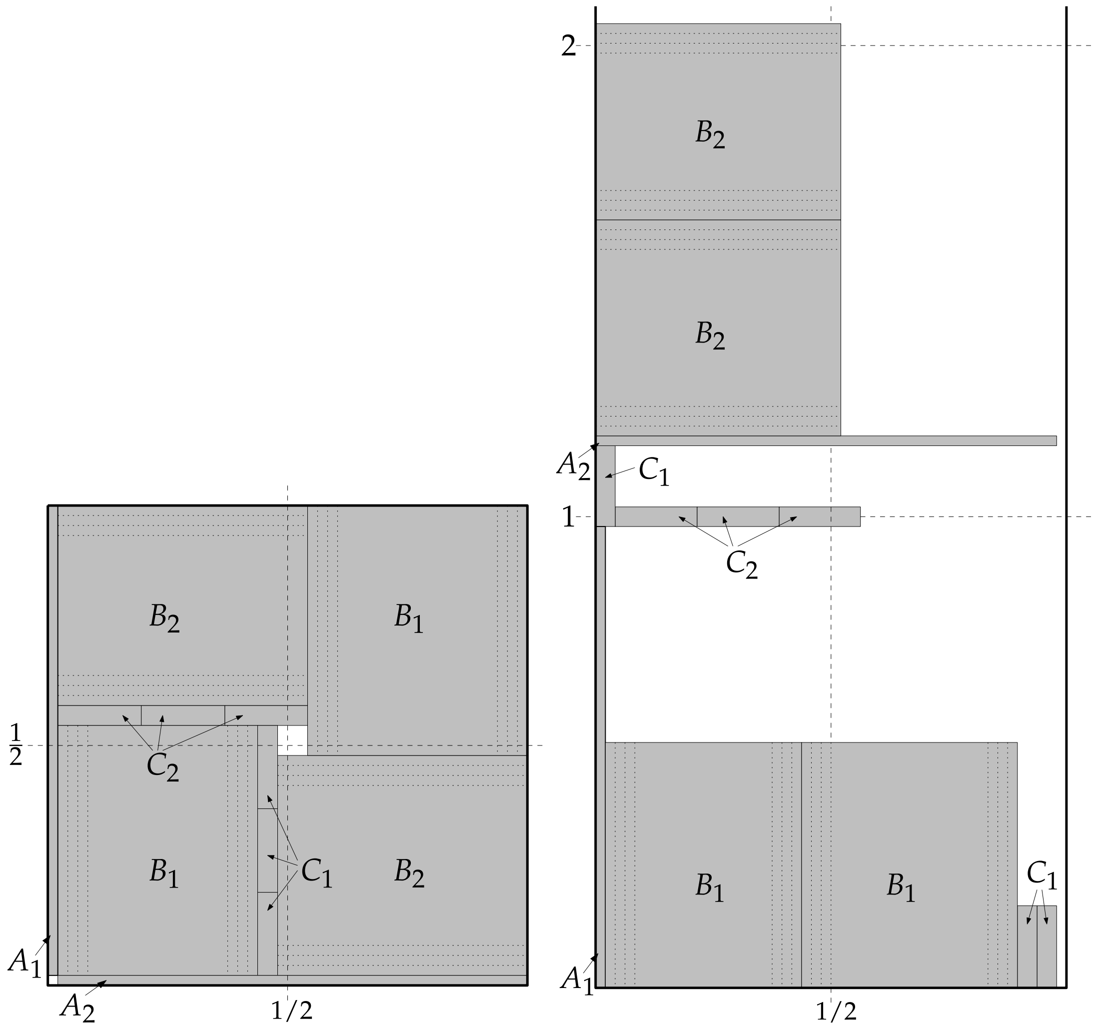


Figure 25: A lower bound instance for HFF. The left side shows that $\mathrm{OPT}(I) = 1$ whereas the right side shows the FFDH packing. FFD packs A_2 together with the first layer into a bin. The remaining layers have total height greater than 1 and thus do not fit into a second bin.

of height. Then FFD packs the item in A_2 together with the first layer into a bin. Since $\delta < 1/34$, the total height of all remaining layers is

$$h = \overbrace{\frac{1}{6} + \frac{1}{3}\delta}^{C_1} + \overbrace{(\frac{1}{\delta} - 6)\delta}^{B_2} = \frac{7}{6} - \frac{17}{3}\delta > 1.$$

Thus 3 bins are needed to pack the resulting layers with FFD.

Note that the instance that we describe does not change under rotation by 90 degrees. Thus Hybrid First Fit by Width outputs the same packing. We refer the reader to [47] to verify that Zhang's algorithm uses 3 bins as well since the total area of all high and small items is $\mathcal{A}(A_1 \cup B_1 \cup C_1 \cup C_2) > 1/2$ for $\delta > 0$.

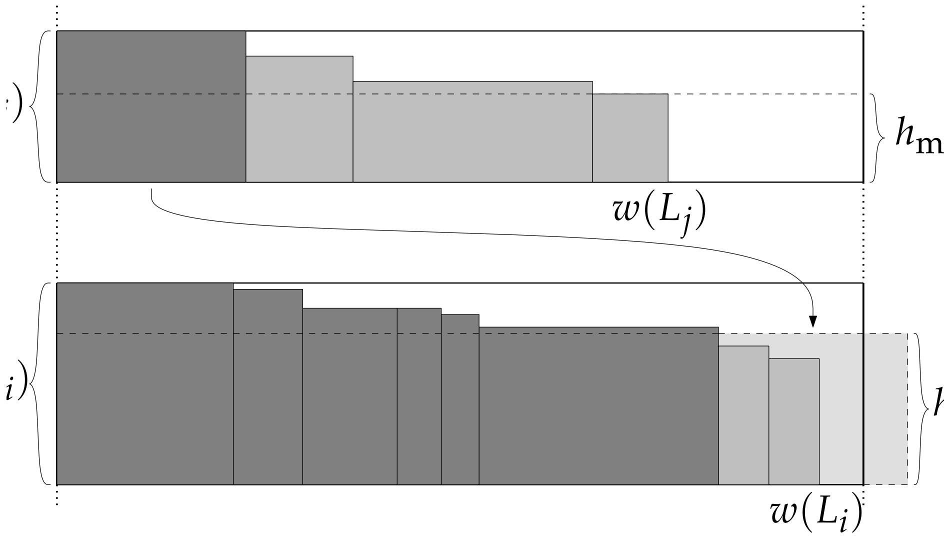


Figure 26: Deriving a bound for the volume in two layers. All items that are packed up to and including the first item in L_j are dark and have height at least $h_{\max}(L_j)$. The total width of these items is greater than 1. All other items have height at least $h_{\min}$.

5.2.3 UPPER BOUND

Before we start with the main proof of the upper bound we introduce the following important lemma.

Lemma 23. *Let L_i, L_j be two layers with $i < j$. Then*

$$\mathcal{A}(L_i \cup L_j) \geq h_{\max}(L_j) + (w(L_i) + w(L_j) - 1)h_{\min},$$

where $h_{\min} = h_{\min}(L_i \cup L_j)$ is the smallest height of the items in $L_i \cup L_j$.

See Figure 26 for an illustration of the following proof.

Proof. First consider all items that are packed up to and including the first item in layer L_j (dark items in Figure 26). These items all have height at least $h_{\max}(L_j)$. Let w' be the total width of these items. We have $w' > 1$ since the first item of layer L_j did not fit into layer L_i. Thus we get a total area of at least $w' \cdot h_{\max}(L_j)$ for these items. Since all items in both layers have height at least $h_{\min}$ the remaining width of $w(L_i) + w(L_j) - w'$ makes up an additional area of at least $(w(L_i) + w(L_j) - w')h_{\min}$. Thus the total area is

$$\begin{aligned}\mathcal{A}(L_i \cup L_j) &\geq w' h_{\max}(L_j) + (w(L_i) + w(L_j) - w')h_{\min} \\ &\geq h_{\max}(L_j) + (w(L_i) + w(L_j) - 1)h_{\min}.\end{aligned}$$

Note that $w(L_i) + w(L_j) - 1 \geq 0$ since otherwise the first item of L_j fits into L_i. □

With the previous lemma we are able to derive bounds for the total area of the items that are packed. In addition to the most intuitive lower bound

$$\mathrm{OPT}(I) \geq \mathcal{A}(I) \tag{5.8}$$

we use the following two bounds. Let S be the set of layers that contain exactly one item. Since it is not possible to pack two of these items next to each other (otherwise it would have been done by the algorithm) the total height of these layers form the lower bound

$$\mathrm{OPT}(I) \geq \sum_{L_i \in S} h_{\max}(L_i). \tag{5.9}$$

Finally, the set $T = \{r_i = (w_i, h_i) \mid h_i > 1/2\}$ of items of height greater than 1/2 provides the last lower bound

$$\mathrm{OPT}(I) \geq \sum_{r_i \in T} w_i \tag{5.10}$$

that we use.

Assume for the sake of contradiction that HFF uses more than $3\,\mathrm{OPT}(I)$ bins. Let L_k be the first layer from FFDH that is packed into bin number $3\,\mathrm{OPT}(I) + 1$ and let $r^* = (w^*, h^*)$ be the first item in L_k. Discard all items that are considered after r^* by FFDH. Note that the packing that remains is exactly the packing that HFF produces on the reduced instance I'. Therefore we argue about this reduced instance in the remainder of this section.

Lemma 24. *We have $h^* \leq 1/3$.*

Proof. Suppose that $h^* > 1/3$. Then all bins contain either one or two layers and are filled up to a height greater than $1 - h^*$. Layers that are alone in a bin have height greater than $1 - h^*$. Thus if Lemma 23 is applied on two such layers, say L_i, L_j $(i < j)$, we get a combined area of at least $h_{\max}(L_j) \geq 1 - h^*$. Applied on both layers of a bin that contains two layers we get a combined area of at least h^*, which is a lower bound for the height of the smaller layer in the bin.

Let m be the number of bins that contain exactly one layer (except the bin that contains r^*). These are layers $L_1, \ldots, L_m$. Then the other $3\,\mathrm{OPT}(I') - m$ bins contain two layers each. Note that $h_{\max}(L_1) \geq \cdots \geq h_{\max}(L_m) > 1/2$ as otherwise a second layer would fit into the bin. Let w'_i be the width of items of height $> 1/2$ in layer L_i. Thus with the lower bound (5.10) we get

$$\mathrm{OPT}(I') \geq \sum_{i=1}^{m} w'_i \geq \sum_{i=1}^{\lfloor m/2 \rfloor} (w'_{2i-1} + w'_{2i}) > \left\lfloor \frac{m}{2} \right\rfloor \tag{5.11}$$

since any two layers have cumulative width $w'_i + w'_j > 1$.

If m is even we get

$$\begin{aligned}
\mathcal{A}(I') &\geq \overbrace{\frac{m}{2}(1 - h^*)}^{\text{bins with one layer}} + \overbrace{(3\,\mathrm{OPT}(I') - m)\, h^*}^{\text{bins with two layers}} \\
&= \frac{m}{2} + \left(3\,\mathrm{OPT}(I') - \frac{3}{2}m\right) h^* \\
&> \frac{m}{2} + \mathrm{OPT}(I') - \frac{m}{2} && \text{using } h^* > 1/3 \text{ and (5.11)} \\
&= \mathrm{OPT}(I').
\end{aligned}$$

For odd values of m, say $m = 2n + 1$, we can apply Lemma 23 on the first $2n$ layers as before and on layer L_m together with layer L_k consisting of r^*. This given another area of h^* and we get

$$\begin{aligned}\mathcal{A}(I') &\geq \overbrace{n\,(1-h^*) + h^*}^{\text{bins with one layer}} + \overbrace{(3\,\mathrm{OPT}(I') - (2n+1))\,h^*}^{\text{bins with two layers}} \\ &= n + (3\,\mathrm{OPT}(I') - 3n)\,h^* \\ &> n + \mathrm{OPT}(I') - n \qquad \text{using } h^* > 1/3 \text{ and (5.11)} \\ &= \mathrm{OPT}(I').\end{aligned}$$

In both cases we get a the contradiction $\mathcal{A}(I') > \mathrm{OPT}(I')$ and thus $h^* \leq 1/3$. □

In the following step we will use Lemma 23 on pairs of consecutive layers in order to derive a lower bound on the total area of the items.

As r^* is packed into a new bin, all previous bins contain layers of total height greater than $1 - h^*$. Thus we get the following bound for the total height h of the first $k - 1$ layers:

$$h = \sum_{i=1}^{k-1} h_{\max}(L_i) > 3\,\mathrm{OPT}(I')\,(1-h^*).$$

We need a slightly different bound, which follows immediately since the first bin contains at least layer L_1 of height $h_{\max}(L_1)$ and all other bins contain layers of total height greater than $1 - h^*$:

$$h = \sum_{i=1}^{k-1} h_{\max}(L_i) > (3\,\mathrm{OPT}(I') - 1)(1-h^*) + h_{\max}(L_1). \tag{5.12}$$

Applying Lemma 23 on pairs of consecutive layers L_{2i-1}, L_{2i} and adding the area of r^* we get

$$\begin{aligned}\mathcal{A}(I') &\geq \sum_{i=1}^{\lfloor (k-1)/2 \rfloor} \mathcal{A}(L_{2i-1} \cup L_{2i}) + w^* h^* \\ &\geq \sum_{i=1}^{\lfloor (k-1)/2 \rfloor} \big(h_{\max}(L_{2i}) + (w(L_{2i-1}) + w(L_{2i}) - 1)h^*\big) + w^* h^* \\ &\qquad \text{by Lemma 23} \\ &\geq \sum_{i=1}^{\lfloor (k-1)/2 \rfloor} h_{\max}(L_{2i}) + \sum_{i=1}^{k-1} \big(w(L_i) - \frac{1}{2}\big)h^* + w^* h^*. \end{aligned} \tag{5.13}$$

We first derive a lower bound for the first part of the previous inequality. With $h_{2i} \geq h_{2i+1}$ we get

$$2 \sum_{i=1}^{\lfloor (k-1)/2 \rfloor} h_{\max}(L_{2i}) \geq \sum_{i=2}^{k-1} h_{\max}(L_i) = h - h_1 \tag{5.14}$$

and thus

$$\begin{aligned}
\sum_{i=1}^{\lfloor (k-1)/2 \rfloor} h_{\max}(L_{2i}) &\geq \frac{h - h_{\max}(L_1)}{2} && \text{by (5.14)} \\
&> \frac{(3\,\mathrm{OPT}(I') - 1)(1 - h^*)}{2} && \text{by (5.12)} \\
&\geq \mathrm{OPT}(I')(1 - h^*) + \frac{\mathrm{OPT}(I') - 1}{2}(1 - h^*) \\
&\geq \mathrm{OPT}(I') - \mathrm{OPT}(I')h^* + \frac{\mathrm{OPT}(I') - 1}{2} 2h^* && \text{as } h^* \leq 1/3 \\
&\geq \mathrm{OPT}(I') - h^*. && (5.15)
\end{aligned}$$

To simplify the presentation let $A = \sum_{i=1}^{k-1} \left(w(L_i) - \frac{1}{2}\right)h^* + w^* h^*$. Inequalities (5.13) and (5.15) lead to the lower bound

$$\mathcal{A}(I') > \mathrm{OPT}(I') - h^* + A.$$

We need the following observation before we can derive a contradiction.

Observation 1. *There are at least 6 layers and if* $\mathrm{OPT}(I') = 1$ *then the three largest layers are packed into the first two bins.*

The observation is obvious for $\mathrm{OPT}(I') > 1$ as we assume that $3\,\mathrm{OPT}(I') + 1$ bins are used. If $\mathrm{OPT}(I') = 1$ then the height of layer L_2 can not be greater than $1/2$ as otherwise in contradiction to (5.10):

$$\sum_{\substack{r_i = (w_i, h_i) \\ h_i > 1/2}} w_i > 1 = \mathrm{OPT}(I').$$

Thus in this case the three largest layers are packed into the first two bins and each but the first bin contains at least 2 layers. We are now ready to prove the following theorem.

Theorem 9. *The approximation ratio of* HYBRID FIRST FIT *is 3.*

Proof. The lower bound was already given in Section 5.2.2. We consider two different cases to derive a contradiction on the assumption that r^* is packed into bin number $3\mathrm{OPT}(I') + 1$. In the first case we show that $A \geq h^*$ which leads to $\mathcal{A}(I') > \mathrm{OPT}(I')$ as a contradiction to the lower bound (5.8). The tricky part in this case is to consider that there might be a layer of width $w(L_i) < 1/2$ which would result in a negative term in the sum of A. But there can be at most one such layer as otherwise both layers would fit together. In the second case we use the lower bound (5.9) that is given by the total height of layers that consist of exactly one item.

CASE 1. Assume that there are 3 or more layers with width at least $2/3$. Let L_u, L_v, L_w be the layers with greatest total width and let L_t be the layer with the smallest total width. Since there are more than 4 layers we can assume w.l.o.g. that $t \neq \{u, v, w\}$. Then

$$\begin{aligned}
A &= \sum_{i=1}^{k-1} \left(w(L_i) - \frac{1}{2}\right)h^* + w^* h^* \\
&\geq 3\left(\frac{2}{3} - \frac{1}{2}\right)h^* + \sum_{\substack{i=1 \\ i \neq t,u,v,w}}^{k-1} \left(w(L_i) - \frac{1}{2}\right)h^* + \left(w(L_t) - \frac{1}{2}\right)h^* + w^* h^* \\
&> \frac{1}{2}h^* + \sum_{\substack{i=1 \\ i \neq t,u,v,w}}^{k-1} \left(w(L_i) - \frac{1}{2}\right)h^* + \frac{1}{2}h^*.
\end{aligned}$$

The last step is due to $w(L_t) > 1 - w^*$ as otherwise r^* fits into layer L_t. Since there is at most one layer with width $w(L_j) < 1/2$ (and this would be L_t), the sum in the middle in non-negative. Thus $A > h^*$, which gives a contradiction to the lower bound (5.8).

CASE 2. Now assume that there are less than 3 layers with width at least $2/3$. Let L_ℓ be the first layer (lowest indexed layer) with width $w(L_\ell) < 2/3$. Consider an item $r_j = (w_j, h_j)$ in layer L_i with $i > \ell$. Then $w_j > 1/3$ since otherwise r_j fits into layer L_ℓ. Thus $w_i > 2/3$ or L_i contains exactly one item. Let S be the set of layers that contain exactly one item and consider the lower bound (5.9).

$$\begin{aligned}
\mathrm{OPT}(I') &\geq \sum_{L_i \in S} h_{\max}(L_i) \\
&\geq \sum_{i=1}^{k} h_{\max}(L_i) - \big(h_{\max}(L_1) + h_{\max}(L_2) + h_{\max}(L_3)\big),
\end{aligned}$$

as there are at most three layers that contain more than one item and these three layers have total height at most $h_{\max}(L_1) + h_{\max}(L_2) + h_{\max}(L_3)$. For $\mathrm{OPT}(I') = 1$, Observation 1 implies the contradiction

$$\mathrm{OPT}(I') \geq \sum_{i=1}^{k} h_{\max}(L_i) - (h_{\max}(L_1) + h_{\max}(L_2) + h_{\max}(L_3)) > 1.$$

For $\mathrm{OPT}(I') \geq 2$, at most 3 bins contain layers with more than one item. Thus we get

$$\begin{aligned}
\mathrm{OPT}(I') &\geq \sum_{L_i \in S} h_{\max}(L_i) \\
&> (3\,\mathrm{OPT}(I') - 3)(1 - h^*) + h^* \\
&\geq \mathrm{OPT}(I').
\end{aligned}$$

The last step follows from $(3\,\mathrm{OPT}(I') - 3)(1 - h^*) + h^* \geq 2\,\mathrm{OPT}(I') - 2$ since $h^* \leq 1/3$. □

Part II

ONLINE ALGORITHMS

6 STRIP PACKING

As mentioned in the introduction, Brown, Baker & Katseff[8] derived a lower bound of 2 on the competitive ratio of strip packing by constructing certain adversary sequences. Just recently, Kern & Paulus[35] derived a matching upper and lower bound of $3/2 + \sqrt{33}/6 \approx 2.457$ for packing these Brown-Baker-Katseff sequences, that we call BBK sequences in the sequel.

Using modified BBK sequences we show an improved lower bound of $2.589\dots$ on the absolute competitive ratio of this problem. The modified sequences that we use consist solely of two types of items, namely, *thin* items that have negligible width (and thus can all be packed in parallel) and *blocking* items that have width 1. The advantage of these sequences is that the structure of the optimal packing is simple, i.e., the optimal packing height is the sum of the heights of the blocking items plus the maximal height of the thin items. Therefore, we call such sequences *primitive*.

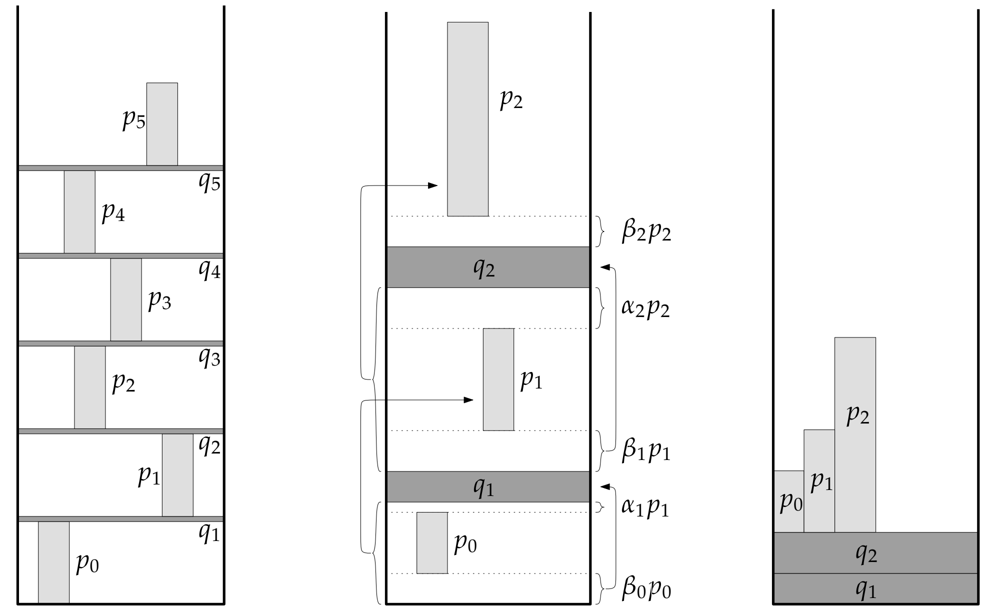
On the positive side, we present an online algorithm for packing primitive sequences with competitive ratio $(3 + \sqrt{5})/2 = 2.618\dots$. This upper bound is especially interesting as it not only applies to the concrete adversary instances that we use to show our lower bound. Thus to show a new lower bound for strip packing that is greater than $2.618\dots$ (and thus reduce the gap to the general upper bound of 6.6623), new techniques are required that take instances with more complex optimal solutions into consideration.

We start with a description of the BBK sequences and their modification in Section 6.1. In Section 6.2 we present our lower bound based on these modifications and in Section 6.3 we give our algorithm for packing primitive sequences. Finally, in Section 6.4 we further discuss the competitive ratio for this restricted problem and present a promising approach to further improve our lower bound.

6.1 SEQUENCE CONSTRUCTION

We denote the thin items by p_i and the blocking items by q_i (adopting the notation from [35]). As already mentioned in the introduction, we assume that the width of the thin items is negligible and thus all thin items can be packed next to each other. Moreover, the width of the blocking items q_i is always 1, so that no item can be packed next to any other item in parallel. Therefore, all items are characterized by their heights and we refer to their heights by p_i and q_i as well. By definition, for any list $L = q_1, q_2, \dots, q_k, p_1, p_1, \dots, p_\ell$ consisting of thin and blocking items we have

$$\mathrm{OPT}(L) = \sum_{i=1}^{k} q_i + \max_{i=1,\dots,\ell} p_i.$$



(a) A greedy packing (b) A packing of a BBK sequence (c) An optimal packing of a BBK sequence

Figure 27: Online and optimal packings

To prove the desired lower bound we assume the existence of a ρ-competitive algorithm ALG for some $\rho < 2.589\ldots$ (the exact value of this bound is specified later) and construct an adversary sequence depending on the packing that ALG generates.

To motivate the construction, let us first consider the GREEDY algorithm for online strip packing, which packs every item as low as possible—see Figure 27a. This algorithm is not competitive (i.e., has unbounded competitive ratio): Indeed, consider the list $L_n = p_0, q_1, p_1, q_2, p_2, \ldots, q_n, p_n$ of items with

$$p_0 := 1,$$
$$q_i := \varepsilon \qquad \text{for } 1 \leq i \leq n,$$
$$p_i := p_{i-1} + \varepsilon \qquad \text{for } 1 \leq i \leq n$$

for some $\varepsilon > 0$. GREEDY would pack each item on top of the preceding ones and thus generate a packing of height $\text{GREEDY}(L_n) = \sum_{i=0}^{n} p_i + \sum_{i=1}^{n} q_i = n + 1 + \mathcal{O}(\varepsilon)$, whereas the optimum clearly has height $1 + \mathcal{O}(\varepsilon)$.

The GREEDY algorithm illustrates that any competitive online algorithm needs to create gaps in the packing. These gaps work as a buffer to accommodate small blocking items—or, viewed another way, force the adversary to release larger blocking items.

BBK SEQUENCES. The idea of Brown, Baker & Katseff[8] was to try to cheat an arbitrary (non-greedy) online packing algorithm ALG in a similar way by constructing an alternating sequence $p_0, q_1, p_1, \ldots$ of thin and blocking items. The heights p_i respectively q_i are determined so as to force the online algorithm ALG to put each item above the previous ones—see Figure 27b for an illustration. To describe the heights of the items formally, we consider the gaps that ALG creates between the items. We distinguish two types of gaps, namely gaps below and gaps above a blocking item, and refer to theses gaps as α- and β-gaps, respectively. These gaps also play an important role in our analysis of the modified BBK sequences. We describe the height of the gaps around the blocking item q_i relative to the thin item p_i. Thus, we denote the height of the α-gap below q_i by $\alpha_i p_i$ and the height of the β-gap above q_i by $\beta_i p_i$. Using this notation, we are ready to formally describe the BBK sequences $L = p_0, q_1, p_1, q_2, \ldots$ with

$$
\begin{aligned}
p_0 &:= 1, \\
q_1 &:= \beta_0 p_0 + \varepsilon, \\
p_i &:= \beta_{i-1} p_{i-1} + p_{i-1} + \alpha_i p_i + \varepsilon && \text{for } i \geq 1, \\
q_i &:= \max\left(\alpha_{i-1} p_{i-1}, \beta_{i-1} p_{i-1}, q_{i-1}\right) + \varepsilon && \text{for } i \geq 2.
\end{aligned}
$$

As mentioned in the introduction, Brown, Baker & Katseff[8] used these sequences to derive a lower bound of 2 before Kern & Paulus[35] recently showed that the competitive ratio for packing them is $\rho_{\text{BBK}} = 3/2 + \sqrt{33}/6 \approx 2.457$.

The optimal online algorithm for BBK sequences that Kern & Paulus[35] describe generates packings with striking properties: No gaps are created except the first possible gap $\beta_0 = \rho_{\text{BBK}} - 1$ and the second α-gap $\alpha_2 = 1/(\rho_{\text{BBK}} - 1)$, which are chosen as large as possible while remaining ρ_{BKK}-competitive. Observing this behavior of the optimal algorithm led us to the modification of the BBK sequences.

MODIFIED BBK SEQUENCES. When packing BBK sequences, a good online algorithm should be eager to enforce blocking items of relatively large size (as each blocking item of size q increases the optimal packing by q as well). These blocking items are enforced by generating corresponding gaps.

Modified BBK sequences are designed to counter this strategy: Each time the online algorithm places a blocking item q_i, the adversary, rather than immediately releasing a thin item p_{i+1} that does not fit in between the last two blocking items, generates a whole sequence of slowly growing thin items, which "continuously" grow from p_i to p_{i+1}. Packing this subsequence causes additional problems for the online algorithm: If the algorithm fits the whole subsequence into the last interval between q_{i-1} and q_i, it would fill out the whole interval and create an α-gap of 0. On the other extreme, if ALG would pack a thin item of height roughly p_i above q_i, then the (relative) β-gap it can generate is much less compared to what it could have achieved with a thin item of larger height p_{i+1}. The next blocking item q_{i+1} will be released as soon as the sequence of thin items has grown from p_i to p_{i+1}.

This general concept of the modified BBK sequences applies after the first blocking item q_1 is released. Since subsequences of thin items and single blocking items are released alternately, we refer to this phase as the *alternating* phase. Before that, we have a *starting* phase which ends with the release of the first blocking item q_1. This starting phase needs special attention as we have no preceding interval height as a reference.

The optimal online algorithm by Kern & Paulus[35] generates an initial gap $\beta_0 = \rho_{\text{BBK}} - 1$ of maximal size to enforce a large first blocking item q_1. In the starting phase, we seek to prevent the algorithm from creating a large β_0-gap in the following way. Assume that the online algorithm places p_0 "too high" (i.e., β_0 is "too large"). Then the adversary, instead of releasing q_1, would continue generating higher and higher thin items and observe how the algorithm places them. As long as the algorithm places these thin items next to each other (overlapping in their packing height), the size of the gap below these items decreases monotonically relative to the height where items are packed. Eventually, β_0 has become sufficiently small—in which case the starting phase comes to an end with the release of q_1—or the online algorithm decides to "jump" in the sense that one of the items in this sequence of increasing height thin items is put strictly above all previously packed thin items, creating a new gap (distance between the last two items) and a significantly increased new packing height. Once a jump has occurred, the adversary continues generating thin items of slowly growing height until a next jump occurs or until the ratio of the largest current gap to the current packing height (the modified analogue to the standard β_0-gap) is sufficiently small and the starting phase comes to an end.

Summarizing, a modified BBK sequence consists of simply a sequence of thin items, continuously growing in height, interleaved with blocking items which (by definition of their height) must be packed above all preceding items. We later describe in more detail when exactly the blocking items are released in the corresponding phase.

In the next section we use these modified BBK sequences to show the following theorem.

Theorem 10. *There exists no algorithm for online strip packing with competitive ratio*

$$\rho < \hat{\rho} = \frac{17}{12} + \frac{1}{48}\sqrt[3]{22\,976 - 768\sqrt{78}} + \frac{1}{12}\sqrt[3]{359 + 12\sqrt{78}} \approx 2.589\ldots.$$

6.2 LOWER BOUND

For the sake of contradiction, we assume that ALG is a ρ-competitive algorithm for online strip packing with $\rho < \hat{\rho}$. Let $\delta = \hat{\rho} - \rho > 0$. W.l.o.g. we assume that δ is sufficiently small.

We distinguish between the thin items p_i (whose height matches the height of the previous interval plus an arbitrarily small excess) and the subsequences of gradually growing thin items by denoting the whole sequence of thin items by $r_1, r_2, \ldots$ and designating certain thin items as p_i.

Our analysis differentiates two phases. In the first phase, that we call the *starting* phase, we consider the following problem that the online algorithm faces. Given an input that consists only of thin items $r_1, r_2, \ldots$ (in this phase no blocking items are released), minimize the competitive ratio while retaining a free gap of maximal size (relative to the current packing height). More specifically, let

$$\frac{h(\text{maxgap}_{\text{ALG}}(r_i))}{\text{ALG}(r_i)}$$

be the *max-gap-to-height* ratio after packing r_i where $h(\text{maxgap}_{\text{ALG}}(r_i))$ denotes the height of the maximal gap that algorithm ALG created up to item r_i and $\text{ALG}(r_i)$

denotes the height algorithm ALG consumed up to item r_i. We say ALG is (ρ, c)-competitive in the starting phase if ALG is ρ-competitive (i.e., $\text{ALG}(r_i) \le \rho\text{OPT}(r_i)$) and retains a max-gap-to-height ratio of c (i.e., $h(\text{maxgap}_{\text{ALG}}(r_i))/\text{ALG}(r_i) \ge c$ for $i \ge 1$) for all lists $L = r_1, r_2, \ldots$ of thin items.

In the analysis of the starting phase in Section 6.2.1 we show that our modified BBK sequences force any ρ-competitive algorithm to reach a state with max-gap-to-height ratio less than

$$\hat{c} = \frac{\hat{\rho} - 2\sqrt{\hat{\rho} - 1}}{\hat{\rho} - 1}.$$

Thus no $(\rho, \hat{c})$-competitive algorithm exists for $\rho < \hat{\rho}$. In the moment ALG packs an item r_i and hereby reaches a max-gap-to-height ratio of less than $\hat{c}$, the starting phase ends with the release of the first blocking item q_1 of height $\hat{c} \cdot \text{ALG}(r_i)$.

In the analysis of the alternating phase in Section 6.2.2 we show that no ρ-competitive algorithm can exist if the first blocking item after the starting phase has height $\hat{c}$ times the current packing height for

$$\hat{c} = \frac{1 - \sqrt{4\hat{\rho}^2 - 12\hat{\rho} + 5}}{2(\hat{\rho} - 1)}.$$

Thus our two phases fit together for

$$\hat{c} = \frac{\hat{\rho} - 2\sqrt{\hat{\rho} - 1}}{\hat{\rho} - 1} = \frac{1 - \sqrt{4\hat{\rho}^2 - 12\hat{\rho} + 5}}{2(\hat{\rho} - 1)},$$

which is satisfied for

$$\hat{\rho} = \frac{17}{12} + \frac{1}{48}\sqrt[3]{22\,976 - 768\sqrt{78}} + \frac{1}{12}\sqrt[3]{359 + 12\sqrt{78}} \approx 2.589\ldots.$$

We get a resulting value of $\hat{c} \approx 0.04275\ldots$.

6.2.1 THE STARTING PHASE

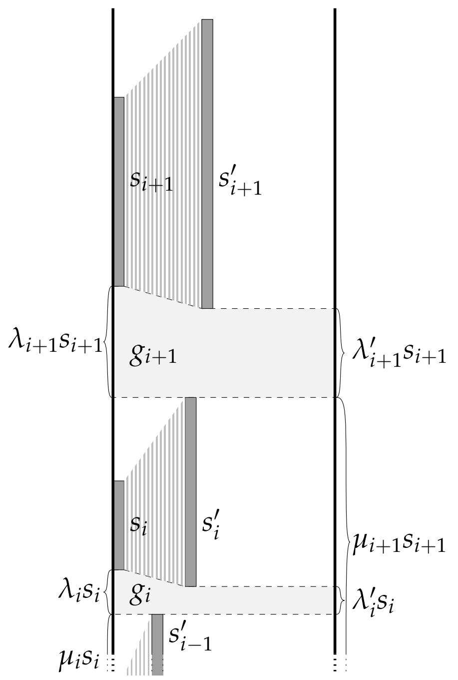
In this section we describe the lower bound for the starting phase. As we explained before, the key parameter of this phase is the *max-gap-to-height* ratio. We will show that for $\rho < \hat{\rho}$, any ρ-competitive algorithm can be forced into a state with max-gap-to-height ratio less than $\hat{c}$. In this section we use the definition

$$\hat{c} = \frac{\hat{\rho} - 2\sqrt{\hat{\rho} - 1}}{\hat{\rho} - 1}.$$

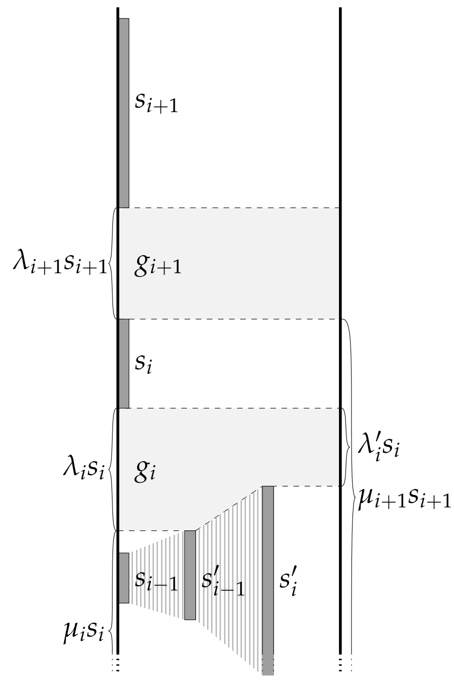
The starting phase is over in the moment a state is reached with a max-gap-to-height ratio of less than $\hat{c}$. To show a contradiction, we assume that the ρ-competitive algorithm ALG is $(\rho, \hat{c})$-competitive, i.e., retains a max-gap-to-height ratio of $\hat{c}$.

Let $\eta > 0$ be some very small constant and consider the adversary list $L_{\text{start}} = r_1, r_2, \ldots$ consisting of thin items

$$r_1 = 1 \qquad \text{and}$$
$$r_i = r_{i-1} + \eta \qquad \text{for } i \ge 2.$$



(a) Notations in the starting phase



(b) Situation when ALG packs s'_i *below* s_i

Figure 28: Starting phase. Lemma 25 shows that the gap sizes are increasing with each jump and Lemma 26 shows that ALG needs to pack s'_i next to s_i

Recall that we denote the thin items by r_i instead of p_i here to be able to designate certain items that correspond to the thin items p_i from the BBK sequence in the analysis of the alternating phase

The sole function of the positive term η is to gradually increase the height of items (we substituted ε from the BBK sequences by η because we use ε later in our analysis). To simplify the calculations, however, we assume that η is chosen small enough such that single instances of η can be omitted from the analysis. (The careful reader might want to check that the bounds we derive for the competitive ratio are actually continuous functions of η and therefore we are well allowed to take the limit ($\eta \to 0$)).

In the following analysis we consider the phases between the creation of new gaps. See Figure 28a for an illustration of the following notations. We refer to the first items in each phase as the *jump* items $s_1, s_2, \dots$ and we denote the last item in each phase by $s'_1, s'_2, \dots$. As we argued above, we assume $s_{i+1} = s'_i$. Furthermore, we denote the gaps that ALG creates by $g_1, g_2, \dots$ and refer to the maximal gap after ALG packs an item r_i by $\text{maxgap}_{\text{ALG}}(r_i)$. Note that the height of the gaps might change when further items are packed (in case ALG packs them such that they reach into the gap from above or below). We denote the initial height of gap g_i by $\lambda_i s_i$ and the gap height directly before the next jump, i.e., in the moment s'_i is packed, by $\lambda'_i s_i$. Note that the height of gap g_i is always given relative to the corresponding jump item s_i. Finally, we denote the packing height up to gap g_i by $\mu_i s_i$, again relative to s_i. We have $\text{ALG}(s_i) = \mu_i s_i + \lambda_i s_i + s_i$ and $\mu_i s_i \geq \text{ALG}(s'_{i-1})$ as s'_{i-1} is packed below g_i but other items might even reach higher than s'_{i-1}.

Since $\mathrm{OPT}(s_i) = s_i$ and $\mathrm{ALG}(s_i) = (\mu_i + \lambda_i + 1)s_i$ we directly have

$$(\mu_i + \lambda_i + 1)s_i \leq \rho\, s_i \qquad \text{for all } i \geq 1$$

$$\text{and thus} \qquad \mu_i + \lambda_i \leq \rho - 1 \qquad \text{for all } i \geq 1. \tag{6.1}$$

Before we are ready to prove that ALG is forced to reach a state with max-gap-to-height ratio less than $\hat{c}$, we have to show some assumptions that we can make on the algorithm ALG. First, we show that we can assume that ALG generates a packing where the gap preceding s_i is the maximal gap until s_{i+1} is packed for all $i \geq 1$. Or, in other words, ALG generates a packing with increasing gap sizes.

Lemma 25. *We can assume that* ALG *generates a packing that satisfies*

$$\mathrm{maxgap}_{\mathrm{ALG}}(r_j) = g_i \qquad \text{for } r_j \in \{s_i, \ldots, s_i'\}.$$

Proof. The intuition of this proof is simple: A new gap g_i that is not maximal (as long as it is the current gap) is unnecessary and can therefore be omitted. We do this by bottom-aligning all items from s_i to r_j with the top of the previous gap.

More formally, let $\mathrm{maxgap}_{\mathrm{ALG}}(r_j) = g_k \neq g_i$ be the first violation of the condition for $r_j \in \{s_i, \ldots, s_i'\}$. The modified algorithm ALG′ simulates ALG with the exception that it bottom-aligns those items from $\{s_i, \ldots, r_j\}$ that were previously packed above g_k with the top of g_k.

As items have only been moved downwards, ALG′ remains ρ-competitive. Moreover, for the altered algorithm we have

$$\mathrm{maxgap}_{\mathrm{ALG}'}(r_\ell) = \mathrm{maxgap}_{\mathrm{ALG}}(r_j) = g_k \qquad \text{for } r_\ell \in \{s_i, \ldots, r_j\}$$

$$\text{and} \qquad h_{r_\ell}(g_k) \geq h_{r_j}(g_k) \geq \hat{c}\,\mathrm{ALG}(r_j) \geq \hat{c}\,\mathrm{ALG}'(r_\ell) \quad \text{for } r_\ell \in \{s_i, \ldots, r_j\}.$$

where $h_{r_\ell}(g_k)$ denotes the height of gap g_k in the moment r_ℓ is packed. The last inequality shows that the altered algorithm retains a max-gap-to-height ratio of $\hat{c}$. So $(\rho, \hat{c})$-competitiveness is not violated.

In total, the altered algorithm ALG′ potentially even saves packing height in comparison with the original algorithm ALG. By induction we can apply this method to all violations of $\mathrm{maxgap}_{\mathrm{ALG}}(r_j) = g_i$. ☐

Now we show that the space *below* a jump item s_i is not large enough to accommodate s_i' before ALG makes the next jump. The implication of this statement is that any $(\rho, \hat{c})$-competitive algorithm needs to place new items next to the current jump item.

Lemma 26. ALG *cannot generate a gap with an item s_{i+1} when the last item s_i' is packed completely below the previous jump item s_i.*

Proof. For the sake of contradiction assume that ALG generates such a gap with item s_{i+1} while the last item s_i' was packed completely below the previous jump item s_i—see Figure 28b. As we will see, the proof of this lemma does not require to consider that ALG retains a max-gap-to-height ratio of $\hat{c}$.

By Inequality (6.1) we have $s_i' \leq (\mu_i + \lambda_i)s_i \leq (\rho - 1)s_i$ as s_i' is packed below s_i. Thus $s_i \geq s_i'/(\rho - 1)$. With our assumption $s_i' = s_{i+1}$ we have

$$\mathrm{ALG}(s_{i+1}) \geq s_i' + s_i + s_{i+1} \geq \left(2 + \frac{1}{\rho - 1}\right)s_{i+1}.$$

The contradiction follows with $\rho < 2.618\ldots$ as

$$\begin{aligned} & 1 > (\rho-2)(\rho-1) \\ \Leftrightarrow \quad & (2+\frac{1}{\rho-1})s_{i+1} > \rho\, s_{i+1} \\ \Rightarrow \quad & \mathrm{ALG}(s_{i+1}) > \rho\, \mathrm{OPT}(s_{i+1}). \end{aligned}$$

□

Lemmas 25 and 26 state that each jump is bigger than the previous jump and that ALG needs to "grow" the items next to the current jump item. This gives us sufficient information about the structure of the online packing to derive a contradiction. More specifically, the next two lemmas show that the relative gap height λ_i is decreasing by a constant in every step, which contradicts the trivial lower bound of $\lambda_i \geq \hat{c}/(1-\hat{c}) \cdot (\mu_i + 1) \geq \hat{c}/(1-\hat{c})$ as $\lambda_i s_i \geq \hat{c}(\mu_i s_i + \lambda_i s_i + s_i)$.

Lemma 27. *We have $\lambda_1 \leq \rho - 1$ and for any $i \geq 1$*

$$\lambda_{i+1} \leq \rho - 2 - \frac{\hat{c}(\rho-1)}{\lambda_i - \hat{c}(\rho-1)}.$$

Proof. The first part, $\lambda_1 \leq \rho - 1$, follows directly from the ρ-competitiveness.

By Lemma 25 we know that $\mathrm{maxgap}_{\mathrm{ALG}}(s'_i) = g_i$. Since ALG preserves a max-gap-to-height ratio of at least $\hat{c}$, we have $\lambda'_i s_i \geq \hat{c}\,\mathrm{ALG}(s'_i)$. Moreover, by Lemma 26 we have $\mathrm{ALG}(s'_i) \geq \mu_i s_i + \lambda'_i s_i + s'_i$ and thus

$$\begin{aligned} & \lambda'_i s_i \geq \hat{c}\,\mathrm{ALG}(s'_i) \geq \hat{c}((\mu_i + \lambda'_i)s_i + s'_i) \\ \Rightarrow \quad & s_{i+1} = s'_i \leq \frac{\lambda'_i s_i - \hat{c}(\mu_i+\lambda'_i)s_i}{\hat{c}}. \end{aligned} \tag{6.2}$$

Now we consider the packing height $\mu_{i+1}s_{i+1}$. We have $\mu_{i+1}s_{i+1} \geq \mathrm{ALG}(s'_i) \geq (\mu_i + \lambda'_i)s_i + s'_i$ and thus

$$\begin{aligned} \mu_{i+1} &\geq (\mu_i + \lambda'_i)\frac{s_i}{s_{i+1}} + \frac{s'_i}{s_{i+1}} \\ &\geq \frac{\hat{c}(\mu_i+\lambda'_i)}{\lambda'_i - \hat{c}(\mu_i+\lambda'_i)} + 1 && \text{by Inequality (6.2)} \\ &\geq \frac{\hat{c}(\rho-1)}{\lambda_i - \hat{c}(\rho-1)} + 1. \end{aligned}$$

The last step holds since

$$\begin{aligned} \frac{\partial}{\partial\lambda'_i}\Big(\frac{\hat{c}(\mu_i+\lambda'_i)}{\lambda'_i - \hat{c}(\mu_i+\lambda'_i)}\Big) &= \frac{\hat{c}(\lambda'_i - \hat{c}(\mu_i+\lambda'_i)) - \hat{c}(\mu_i+\lambda'_i)(1-\hat{c})}{(\lambda'_i - \hat{c}(\mu_i+\lambda'_i))^2} \\ &= \frac{-\hat{c}\,\mu_i}{(\lambda'_i - \hat{c}(\mu_i+\lambda'_i))^2} < 0 && \text{as } \mu_i > 0 \end{aligned}$$

and thus $\frac{\hat{c}(\mu_i+\lambda'_i)}{\lambda'_i - \hat{c}(\mu_i+\lambda'_i)}$ is minimal for λ'_i maximal, which is $\lambda'_i = \lambda_i = \rho - 1 - \mu_i$ by Inequality (6.1).

Using this lower bound for μ_{i+1} we get

$$\begin{aligned} \lambda_{i+1} &\leq \rho - 1 - \mu_{i+1} && \text{by Inequality (6.1) for } i+1 \\ &\leq \rho - 2 - \frac{\hat{c}(\rho-1)}{\lambda_i - \hat{c}(\rho-1)}. \end{aligned}$$

□

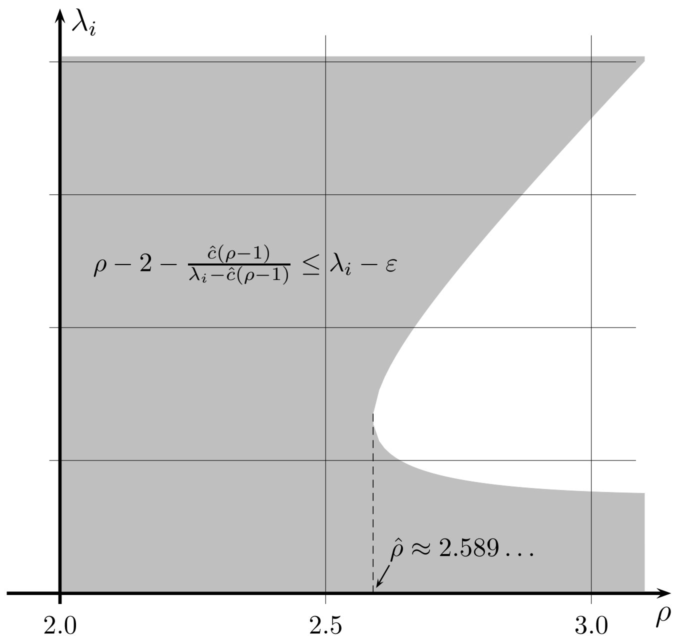


Figure 29: In the shaded area we have $\lambda_{i+1} \leq \lambda_i - \varepsilon$, i.e., the relative gap heights gradually decrease over time

Using this upper bound for the relative gap height λ_{i+1} we will show that no $(\rho, \hat{c})$-competitive algorithm exists. We already gave the lower bound of $\lambda_i \geq \hat{c}/(1-\hat{c})$. On the other hand, the following lemma shows that the relative gap heights are gradually decreasing over time. This gives a contradiction to the assumption that ALG can retain a max-gap-to-height ratio of $\hat{c}$. Thus ALG is either not ρ-competitive or we reach a state with a max-gap-to-height ratio of less than $\hat{c}$, which ends the starting phase.

See Figure 29 for an illustration of the condition in which the relative gap heights λ_i are decreasing by a constant in every step.

Lemma 28. *For some fixed $\varepsilon > 0$ we have*

$$\lambda_{i+1} \leq \lambda_i - \varepsilon.$$

Proof. Let $\varepsilon = \varepsilon(\rho) = 2\sqrt{\hat{c}(\rho-1)} - \rho + 2 + \hat{c}(\rho-1)$. By Lemma 27 we have $\lambda_{i+1} \leq \lambda_i - \varepsilon$ since

$$\begin{aligned}
&\rho - 2 - \frac{\hat{c}(\rho-1)}{\lambda_i - \hat{c}(\rho-1)} \leq \lambda_i - 2\sqrt{\hat{c}(\rho-1)} + \rho - 2 - \hat{c}(\rho-1) \\
\Leftrightarrow \quad & \lambda_i^2 - \left(2\sqrt{\hat{c}(\rho-1)} + 2\hat{c}(\rho-1)\right)\lambda_i \geq -\hat{c}(\rho-1) - 2\sqrt{\hat{c}(\rho-1)}\hat{c}(\rho-1) \\
& \qquad - \hat{c}^2(\rho-1)^2 \\
\Leftrightarrow \quad & \left(\lambda_i - \left(\sqrt{\hat{c}(\rho-1)} + \hat{c}(\rho-1)\right)\right)^2 \geq \left(\sqrt{\hat{c}(\rho-1)} + \hat{c}(\rho-1)\right)^2 - \hat{c}(\rho-1) \\
& \qquad - 2\sqrt{\hat{c}(\rho-1)}\hat{c}(\rho-1) - \hat{c}^2(\rho-1)^2 = 0.
\end{aligned}$$

Thus it only remains to show that $\varepsilon(\rho) > 0$.

With $\hat{c} = \frac{\hat{\rho}-2\sqrt{\hat{\rho}-1}}{\hat{\rho}-1}$ we have $\varepsilon(\hat{\rho}) = 0$ since

$$\varepsilon(\hat{\rho}) = 2\sqrt{\hat{c}(\hat{\rho}-1)} - \hat{\rho} + 2 + \hat{c}(\hat{\rho}-1) = 2\sqrt{\hat{\rho} - 2\sqrt{\hat{\rho}-1}} - \hat{\rho} + 2 + \hat{\rho} - 2\sqrt{\hat{\rho}-1}$$

and

$$\begin{aligned} & 2\sqrt{\hat{\rho} - 2\sqrt{\hat{\rho}-1}} - \hat{\rho} + 2 + \hat{\rho} - 2\sqrt{\hat{\rho}-1} = 0 \\ \Leftrightarrow \quad & 2\sqrt{\hat{\rho} - 2\sqrt{\hat{\rho}-1}} = 2\sqrt{\hat{\rho}-1} - 2 \\ \Leftarrow \quad & 4\big(\hat{\rho} - 2\sqrt{\hat{\rho}-1}\big) = 4(\hat{\rho}-1) - 8\sqrt{\hat{\rho}-1} + 4. \end{aligned}$$

Note that this calculation actually defines the lower bound of $\frac{\hat{\rho}-2\sqrt{\hat{\rho}-1}}{\hat{\rho}-1}$ for $\hat{c}$. Now observe that $\hat{c} = \frac{\hat{\rho}-2\sqrt{\hat{\rho}-1}}{\hat{\rho}-1}$ does not depend on ρ and thus we have

$$\frac{\partial}{\partial \rho}\Big(2\sqrt{\hat{c}(\rho-1)} - \rho + 2 + \hat{c}(\rho-1)\Big) = \frac{\hat{c}}{\sqrt{\hat{c}(\rho-1)}} - 1 + \hat{c}.$$

This derivative is negative as

$$\begin{aligned} & \frac{\hat{c}}{\sqrt{\hat{c}(\rho-1)}} < 1 - \hat{c} \\ \Leftarrow \quad & \frac{\hat{c}}{\rho-1} < (1-\hat{c})^2 \\ \Leftrightarrow \quad & \frac{\hat{\rho}-2\sqrt{\hat{\rho}-1}}{(\rho-1)(\hat{\rho}-1)} < \frac{\big(\hat{\rho}-1-\hat{\rho}+2\sqrt{\hat{\rho}-1}\big)^2}{(\hat{\rho}-1)^2} && \text{by definition of } \hat{c} \\ \Leftrightarrow \quad & \hat{\rho} - 2\sqrt{\hat{\rho}-1} < \big(2\sqrt{\hat{\rho}-1}-1\big)^2 \cdot \frac{\rho-1}{\hat{\rho}-1} \\ \Leftarrow \quad & \hat{\rho} - 2\sqrt{\hat{\rho}-1} < \frac{4(\hat{\rho}-1) - 4\sqrt{\hat{\rho}-1}+1}{2} && \text{as } \frac{\rho-1}{\hat{\rho}-1} > \frac{1}{2} \text{ for } \rho \geq 2 \\ \Leftrightarrow \quad & \frac{3}{2} < \hat{\rho}. \end{aligned}$$

Thus $\varepsilon(\rho)$ is strictly decreasing with respect to ρ and $\varepsilon(\hat{\rho}) = 0$. Hence $\varepsilon(\rho) > 0$ for $\rho < \hat{\rho}$ and the lemma follows. □

The previous lemma and the lower bound of $\lambda_i \geq \hat{c}/(1-\hat{c})$ for any $i \geq 1$ together establish the contradiction. We proved the following lemma.

Lemma 29. *Any ρ-competitive algorithm* ALG *can be forced to reach a state where the max-gap-to-height ratio is less than*

$$\hat{c} = \frac{\hat{\rho}-2\sqrt{\hat{\rho}-1}}{\hat{\rho}-1}.$$

6.2.2 the alternating phase

In this section we describe the lower bound in the alternating phase. In this phase we use that by Lemma 29, any ρ-competitive algorithm ALG is forced to reach a state where the max-gap-to-height ratio is less than $\hat{c} = \frac{\hat{\rho}-2\sqrt{\hat{\rho}-1}}{\hat{\rho}-1}$. Now we use the second part of the definition of $\hat{c}$, namely

$$\hat{c} = \frac{1-\sqrt{4\hat{\rho}^2-12\hat{\rho}+5}}{2(\hat{\rho}-1)}.$$

Together, these definitions give us the actual values $\hat{\rho} \approx 2.589\ldots$ and $\hat{c} \approx 0.04275\ldots$.

Our adversary sequence in this phase starts with a first blocking item q_1 and then continues with the list of thin items of gradually increasing height from the starting phase interleaved with further blocking items. Let $\eta > 0$ be some very small constant and let r_k be the last item that was released in the starting phase. Then we continue with the list $L_{\text{alternating}} = q_1, r_{k+1}, r_{k+2}, \ldots$ where

$$q_1 = \hat{c} \cdot \mathrm{ALG}(r_k) \qquad \text{and}$$
$$r_i = r_{i-1} + \eta \qquad \text{for } i \geq k+1.$$

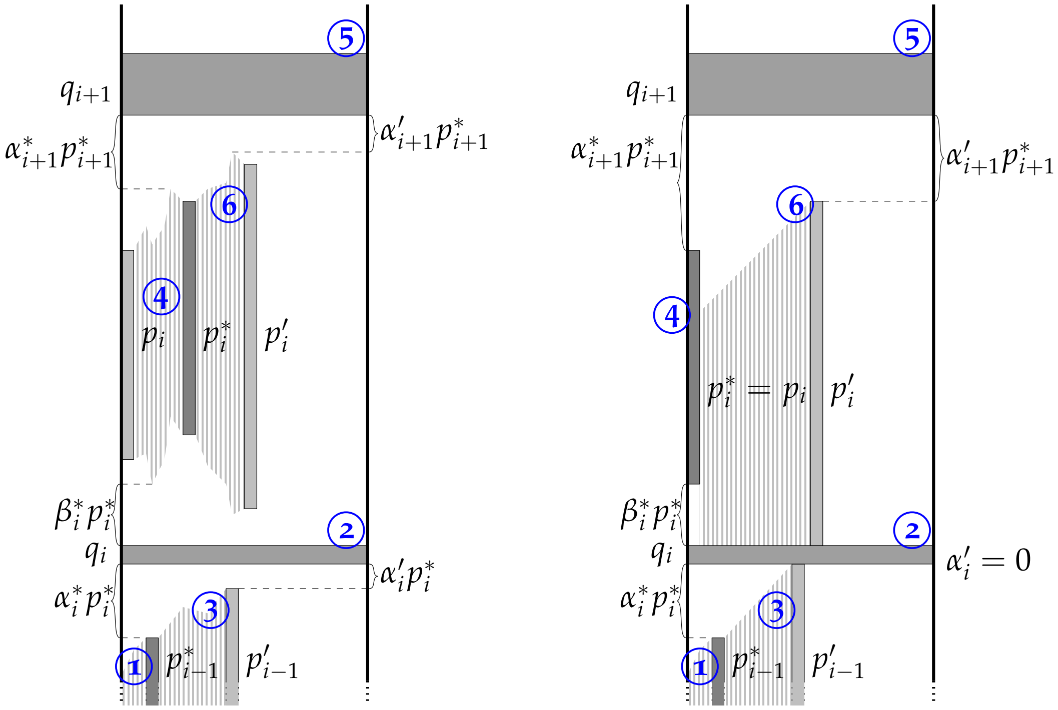
We cannot a priori describe when the further blocking items are inserted into this list as this depends on the packing of the online algorithm. To understand when the blocking items are inserted, let us first introduce the notations in this phase—see Figure 30a.

Similar to the starting phase, we consider the jump items, i.e., the thin items that are the first to be packed above a blocking item q_i, and denote them by p_i. The thin item directly before the jump item is denoted by p'_{i-1} (we will later see that we can actually assume that p'_{i-1} is the last item that is packed below q_i). We denote the interval between the blocking items q_{i-1} and q_i by I_i. As in the standard BBK sequences, the thin item whose height exceeds the height of the previous interval plays an important role. We denote the first item that exceeds the height of I_{i-1} by p^*_i and give all further heights relative to these designated items.

As described in the introduction, we distinguish α-gaps (directly below blocking items) and β-gaps (directly above blocking items). As the gap heights can change during the packing (as further thin items are packed into the same interval) we have to be specific about the moment in which we consider these heights. Let $\alpha^*_i p^*_i$ be the height of the α-gap below q_i in the moment q_i is packed and let $\alpha'_i p^*_i$ be the final height of the α-gap below q_i, i.e., the height in the moment p^*_i is packed (as afterwards no further item can be packed into I_{i-1}). The notation is due to our assumption that p'_{i-1} is the last item that is packed into I_{i-1} (which we show later). Regarding the β-gap we get along with a single definition: Let $\beta^*_i p^*_i$ be the height of the β-gap above q_i in the moment p^*_i is packed.

The blocking item q_{i+1} is released directly after p^*_i. This ensures that the online algorithm jumps before a new blocking item is released (as the height of p^*_i exceeds the height of the previous interval). We set the height of the blocking items to

$$q_1 := \hat{c} \cdot \mathrm{ALG}(r_k) \qquad \text{as already mentioned above and}$$
$$q_i := \max\left(\alpha'_{i-1}p^*_{i-1}, \beta^*_{i-1}p^*_{i-1}, q_{i-1}\right) + \eta \qquad \text{for } i \geq 2.$$



(a) Notations. α-gaps below blocking items with height $\alpha^*_i p^*_i$ in the moment q_i is packed and final height $\alpha'_i p^*_i$. β-gaps above blocking items with height $\beta^*_i p^*_i$

(b) Structured packing according to the assumptions by Lemmas 30, 31. Here we illustrate the case $\beta^*_i > 0$ and thus $\alpha'_i = 0$ and $p^*_i = p_i$

Figure 30: Order of the released items. 1) thin items up to p^*_{i-1}; 2) blocking item q_i; 3) and 4) thin items up to p^*_i (including the jump item p_i); 5) blocking item q_{i+1} and 6) further thin items up to p'_i

Note that we use the *final* height $\alpha'_i p^*_i$ of the α-gap in this definition. This definition ensures that the blocking items are always packed above all previous items.

Again the function of the positive term η is to gradually increase the height of thin items and to ensure that the blocking items are always packed above all previous items. As before, we make the assumption that η is chosen small enough to be omitted from the analysis. Thus we assume that $q_i = \max(\alpha'_{i-1}p^*_{i-1}, \beta^*_{i-1}p^*_{i-1}, q_{i-1})$ and that the height of p^*_i equals the height of the previous interval I_{i-1} throughout this section. (Again, this is justified by taking the limit $(\eta \to 0)$).

We use $\mathrm{succ}(r_i)$ and $\mathrm{prec}(r_i)$ to denote the thin item that succeeds and that precedes r_i, respectively. Using this notations we can rephrase the input list including the blocking items to

$$L_{\text{alternating}} = q_1, r_{k+1}, \ldots, p^*_1, q_1, \mathrm{succ}(p^*_1), \ldots, p^*_2, q_2, \mathrm{succ}(p^*_2), \ldots.$$

We also refer to Figure 30 for an illustration of the order in which the items are released.

OVERVIEW We prove by contradiction that no ρ-competitive algorithm exists for $\rho < \hat{\rho}$. Thus we assume to the contrary that a ρ-competitive algorithm ALG exists. By the analysis of the starting phase we already know that we can force ALG to reach a state with a max-gap-to-height ratio less than $\hat{c}$. In accordance with the notation given above we introduce the parameter γ_i^* to measure how much ALG improves upon the ρ-competitiveness. Let γ_i^* be defined through

$$\mathrm{ALG}(p_i^*) + \gamma_i^* p_i^* = \rho\,\mathrm{OPT}(p_i^*).$$

Using this value, we introduce the potential function

$$\Phi_i = \gamma_i^* + \beta_i^*.$$

Obviously, any ρ-competitive algorithm needs to keep Φ_i non-negative over time. We show the contradiction that Φ_i decreases by a constant in every step. Unfortunately, there is one possible exception to this rule, making the proof substantially more involved: Φ_i might increase exactly once. We will show that even in this case, Φ_i is properly bounded from above and cannot increase a second time.

In a first step we show some valid assumptions on the structure of a packing generated by ALG. Using these assumptions, we enter into the involved induction.

PRELIMINARIES With the next lemmas we investigate some assumptions on the structure of the packing that ALG generates in this phase—see Figure 30b for an illustration. The underlying idea is that if a ρ-competitive algorithm exists, then there also exists a ρ-competitive algorithm that generates packings with the assumed structure.

If the algorithm ALG does not generate such a packing, we can alter the packing (or rather the algorithm) such that the conditions are satisfied and ρ-competitiveness is not violated at any point.

Lemma 30. *We can assume that* ALG *generates a packing such that*

30.1. the items $p_i, \ldots, p_i^*, \ldots, p_i'$ lie in interval I_i,

30.2. the items $p_i, \ldots, p_i^*$ are bottom-aligned,

30.3. the items $\mathrm{succ}(p_i^*), \ldots, p_i'$ are bottom-aligned at the top of q_i.

Proof. By definition the items $p_i^*, \ldots, p_i'$ are taller than the previous interval I_{i-1} and thus all lie in interval I_i. Assume that an item from $p_i, \ldots, \mathrm{prec}(p_i^*)$ does not lie in interval I_i and let r_j be the tallest such item. Then we can move down the items $p_i, \ldots, \mathrm{prec}(r_j)$ and bottom-align them with r_j. This redefines p_i to $\mathrm{succ}(r_j)$ and hereby satisfies Condition 30.1. Observe that moving down the items $p_i, \ldots, \mathrm{proc}(r_j)$ does not violate ρ-competitiveness and as α_i' and β_i^* are not changed, the further packing remains unchanged.

If the items $p_i, \ldots, p_i^*$ are not packed bottom-aligned, we move them downwards until they are aligned with the lowest item of this list in order to satisfy Condition 30.2. And to satisfy Condition 30.3 we move the items $\mathrm{succ}(p_i^*), \ldots, p_i'$ down until they are aligned with the top of q_i if these items are not bottom-aligned at the top of q_i. In both cases the alteration is possible as the height of the interval I_i and thus the height of p_{i+1}^* remains unchanged. Moreover, the height of q_i does not change (as β_i^* is not changed). The values of α_{i+1}^* and α_{i+1}' can actually change, but only become larger.

But as the heights of I_i and p^*_{i+1} remain unchanged, the parameter α'_{i+1} only affects q_{i+1} and the value of q_{i+1} attributes to the packing height of OPT and ALG to the same extent. Thus increased values of α^*_{i+1} and α'_{i+1} cannot cause a violation of the ρ-competitiveness. □

Recall that $q_{i+1} = \max(\alpha'_i p^*_i, \beta^*_i p^*_i, q_i)$. Depending on the way that q_{i+1} is actually defined, we can assume that the other value(s) are zero as the following lemma shows.

Lemma 31. *We can assume that* ALG *generates a packing such that*

31.1. if $q_{i+1} = \max(\beta^*_i p^*_i, q_i)$*, then we have* $\alpha'_i = 0$*,*

31.2. if $q_{i+1} = \max(\alpha'_i p^*_i, q_i)$*, then we have* $\beta^*_i = 0$*.*

Proof. First, assume that $q_{i+1} = \max(\beta^*_i p^*_i, q_i)$ and $\alpha'_i > 0$. By construction of the adversary sequence, the height of p^*_i does not depend on α'_i and is predetermined at the moment q_i is packed. Thus a reduction of α'_i, which corresponds to packing further thin items into the previous interval, does not change q_{i+1} and p^*_i. So we can alter ALG such that all items from $\mathrm{succ}(p'_{i-1}), \ldots, \mathrm{pre}(p^*_i)$, are packed into I_{i-1}. This reduces α'_i to 0 and thus satisfies Condition 31.1 without implying any change to the packing after p^*_i.

Now assume that $q_{i+1} = \max(\alpha'_i p^*_i, q_i)$ and $\beta^*_i > 0$. In this case a reduction of β^*_i does not change q_{i+1} and p^*_i. So we can alter ALG to set β^*_i to 0, i.e., bottom-align the items $p_i, \ldots, p^*_i$ with the top of q_i, without implying any change to the packing after p^*_i and hereby satisfy Condition 31.2. This alteration increases α^*_{i+1} and might increase α'_{i+1} as well—as we saw in Lemma 30, this does not violate ρ-competitiveness. □

Observe that with Lemma 30 we have $p^*_{i+1} = \beta^*_i p^*_i + p^*_i + \alpha^*_{i+1} p^*_{i+1}$ and thus

$$p^*_{i+1} = \frac{1 + \beta^*_i}{1 - \alpha^*_{i+1}} p^*_i. \tag{6.3}$$

Using this equation, we are ready to show the following assumption.

Lemma 32. *We can assume that* ALG *generates a packing such that if* $\alpha'_i > 0$*, then we have*

$$\alpha^*_{i+1} > \frac{(\rho - 1)\alpha'_i}{1 + (\rho - 1)\alpha'_i}.$$

Proof. We assume that $\alpha'_i > 0$ and $\alpha^*_{i+1} \leq (\rho - 1)\alpha'_i / (1 + (\rho + 1)\alpha'_i)$. By Lemma 31 Condition 31.2 we have $\beta^*_i = 0$ and thus $p^*_{i+1} = p^*_i / (1 - \alpha^*_{i+1})$. We can alter ALG to save a packing height of $\alpha'_i p^*_i$ without violating ρ-competitiveness by changing the α-gap to a β-gap. To do that, we move down q_i and all items that are released after p'_{i-1} with the exception of p_i by $\alpha'_i p^*_i$. In other words, we close the $\alpha'_i p^*_i$ gap between p'_{i-1} and q_i by moving down q_i and all items above q_i. The only exception is the item p_i that we keep at its position to retain a β-gap at the moment this item is packed. Hereby, we keep a gap of the original size $\alpha'_i p^*_i$ *above* q_i. See Figure 31 for an illustration of the altered packing.

Note that this alteration changes the adversary sequence: As there does not remain any α'_i-gap, the item q_{i+1} is released directly after p_i is packed—also redefining p^*_i to p_i.

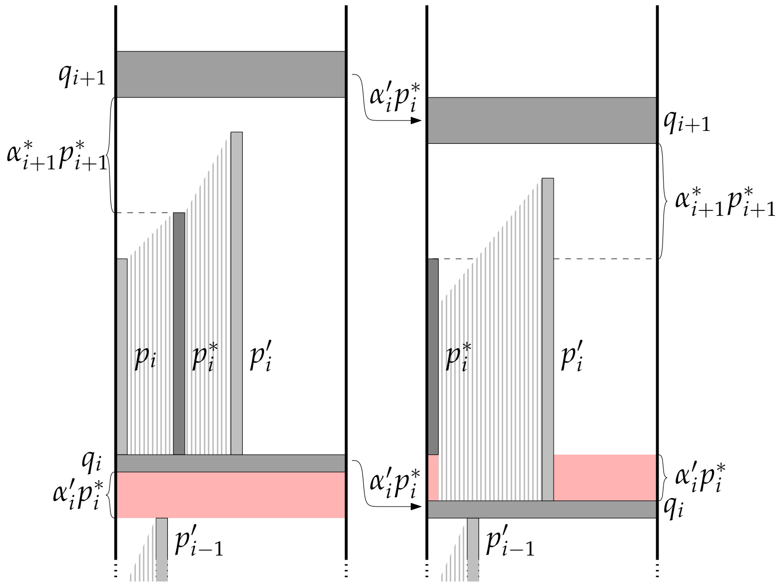


Figure 31: If $\alpha^*_{i+1} \leq (\rho-1)\alpha'_i/(1+(\rho-1)\alpha'_i)$, then we can move down q_i and all items that are released after p'_{i-1} with the exception of p_i by $\alpha'_i p^*_i$. Hereby the α-gap becomes a β-gap and p_i becomes the new p^*_i as the interval I_{i-1} shrinks

This is the only change in the adversary sequence since the size of q_{i+1} is not changed and also the height of interval I_i stays constant. Since the optimal value changed as q_{i+1} is released earlier than before, we have to check whether the altered packing is actually feasible.

We denote the optimal algorithm for the altered instance by OPT′ and the altered algorithm by ALG′. With $\alpha'_i p^*_i$ we refer to the height *before* the alteration. The height $\alpha^*_{i+1}p^*_{i+1}$ remains unchanged. We have $\mathrm{OPT}'(q_{i+1}) = \mathrm{OPT}(p_i) + q_{i+1} = \mathrm{OPT}(p_i) + \alpha'_i p^*_i$ and $\mathrm{ALG}'(q_{i+1}) = \mathrm{ALG}(p_i) + \alpha^*_{i+1}p^*_{i+1} + q_{i+1} = \mathrm{ALG}(p_i) + \alpha^*_{i+1}p^*_{i+1} + \alpha'_i p^*_i$. Thus

$$\begin{aligned} & \mathrm{ALG}'(q_{i+1}) \leq \rho\,\mathrm{OPT}'(q_{i+1}) \\ \Leftrightarrow \quad & \alpha^*_{i+1}p^*_{i+1} \leq \underbrace{\rho\,\mathrm{OPT}(p_i) - \mathrm{ALG}(p_i)}_{\geq 0} + (\rho-1)\alpha'_i p^*_i \\ \Leftarrow \quad & \alpha^*_{i+1} \leq \frac{(\rho-1)\alpha'_i}{1+(\rho-1)\alpha'_i} \qquad \text{by Equation (6.3).} \end{aligned}$$

Thus q_{i+1} can actually be packed by the altered algorithm. The feasibility for all other items in the altered packing is obvious. □

On the other hand, it is not possible for ALG to create an arbitrarily large gap when packing a blocking item q_{i+1}. We capture this fact in the following lemma.

Lemma 33. *We have*

$$\alpha^*_{i+1} \leq \frac{\gamma^*_i + (\rho-1)\frac{q_{i+1}}{p^*_i}}{1+\beta^*_i + \gamma^*_i + (\rho-1)\frac{q_{i+1}}{p^*_i}}.$$

Proof. The value of α^*_{i+1} can be bounded by observing the moment that q_{i+1} is packed. We have

$$\begin{aligned}\mathrm{OPT}(q_{i+1}) &= \mathrm{OPT}(p^*_i) + q_{i+1}\\ \mathrm{ALG}(q_{i+1}) &= \mathrm{ALG}(p^*_i) + \alpha^*_{i+1}p^*_{i+1} + q_{i+1}.\end{aligned}$$

And since q_{i+1} needs to be packed ρ-competitively by ALG we get

$$\begin{aligned}&\mathrm{ALG}(q_{i+1}) \le \rho\,\mathrm{OPT}(q_{i+1})\\ \Leftrightarrow\quad & \alpha^*_{i+1}p^*_{i+1} \le \gamma^*_i p^*_i + (\rho-1)q_{i+1}\\ \Leftrightarrow\quad & \frac{\alpha^*_{i+1}}{1-\alpha^*_{i+1}} \le \frac{\gamma^*_i + (\rho-1)\frac{q_{i+1}}{p^*_i}}{1+\beta^*_i} && \text{by Equation (6.3)}\\ \Leftrightarrow\quad & \alpha^*_{i+1} \le \frac{\gamma^*_i + (\rho-1)\frac{q_{i+1}}{p^*_i}}{1+\beta^*_i+\gamma^*_i+(\rho-1)\frac{q_{i+1}}{p^*_i}}.\end{aligned}$$

□

The parameter α'_i plays an important role in the analysis as the height of the preceding blocking item depends on it. With the next lemma we get an upper bound for this parameter.

Lemma 34. *We have* $(\rho-1)\alpha'_i \le \gamma^*_i$.

Proof. The idea of the bound is that if ALG jumps early, i.e., with an $\alpha'_i > 0$, then it generates a packing where $p^*_i = p'_i + \alpha'_i p^*_i$. This additional height directly contributes to the value of γ^*_i with a factor of $\rho - 1$ (as ALG and OPT increase by the same amount).

Formally, we have $\mathrm{ALG}(p^*_i) = \mathrm{ALG}(p_i) + \alpha'_i p^*_i$, $\mathrm{OPT}(p^*_i) = \mathrm{OPT}(p_i) + \alpha'_i p^*_i$ and $\mathrm{ALG}(p^*_i) + \gamma^*_i p^*_i = \rho\,\mathrm{OPT}(p^*_i)$. And since p_i was feasible we have $\mathrm{ALG}(p_i) \le \rho\,\mathrm{OPT}(p_i)$ and get $(\rho-1)\alpha_i p^*_i \le \gamma^*_i p^*_i$. □

Similar to Kern & Paulus[35] we get the following lemma that bounds the potential function in terms of the parameters of the previous interval.

Lemma 35. *We have*

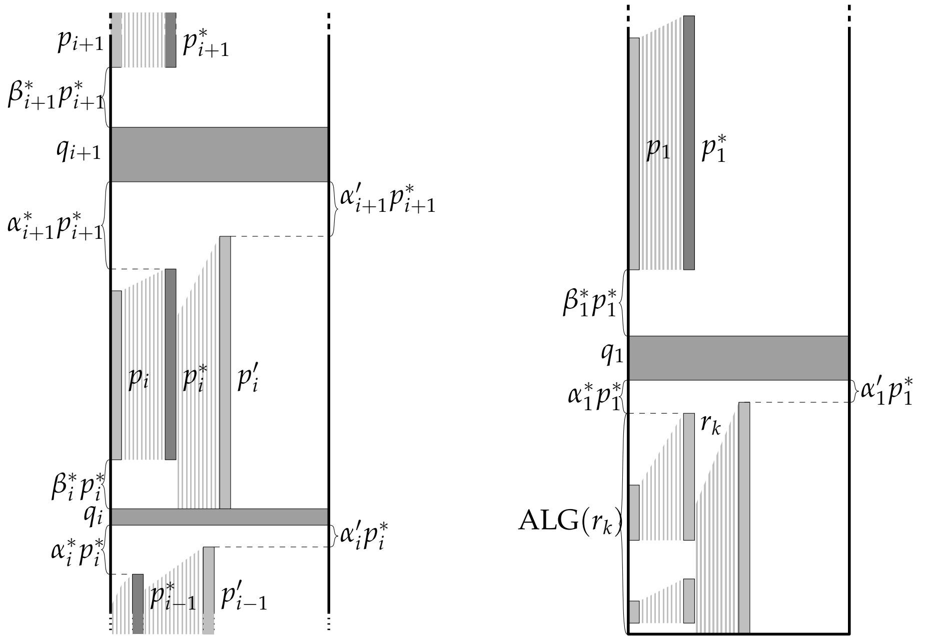
$$\Phi_{i+1} = \gamma^*_{i+1} + \beta^*_{i+1} = \frac{\gamma^*_i + (\rho-1)\frac{q_{i+1}}{p^*_i} + (\rho-1)\beta^*_i - 1}{1+\beta^*_i}(1-\alpha^*_{i+1}) + (\rho-2)\alpha^*_{i+1}.$$

Proof. See Figure 32a for an illustration of the packing. We consider the change between p^*_i and p^*_{i+1} and with $p^*_{i+1} = \beta^*_i p^*_i + p^*_i + \alpha^*_{i+1}p^*_{i+1}$ from Equation (6.3) we have

$$\begin{aligned}\mathrm{OPT}(p^*_{i+1}) &= \mathrm{OPT}(p^*_i) + q_{i+1} + p^*_{i+1} - p^*_i\\ &= \mathrm{OPT}(p^*_i) + q_{i+1} + \beta^*_i p^*_i + \alpha^*_{i+1}p^*_{i+1}\\ \mathrm{ALG}(p^*_{i+1}) &= \mathrm{ALG}(p^*_i) + \alpha^*_{i+1}p^*_{i+1} + q_{i+1} + \beta^*_{i+1}p^*_{i+1} + p^*_{i+1}\\ &= \mathrm{ALG}(p^*_i) + \alpha^*_{i+1}p^*_{i+1} + q_{i+1} + \beta^*_{i+1}p^*_{i+1} + \beta^*_i p^*_i + p^*_i + \alpha^*_{i+1}p^*_{i+1}.\end{aligned}$$

Thus with $\gamma^*_i p^*_i = \rho\,\mathrm{OPT}(p^*_i) - \mathrm{ALG}(p^*_i)$ we get

$$\begin{aligned}&\mathrm{ALG}(p^*_{i+1}) + \gamma^*_{i+1}p^*_{i+1} = \rho\,\mathrm{OPT}(p^*_{i+1})\\ \Leftrightarrow\quad & \gamma^*_{i+1}p^*_{i+1} + \beta^*_{i+1}p^*_{i+1} - (\rho-2)\alpha^*_{i+1}p^*_{i+1} =\\ &\qquad \gamma^*_i p^*_i + (\rho-1)q_{i+1} + (\rho-1)\beta^*_i p^*_i - p^*_i.\end{aligned}$$



(a) In this illustration we disregard Lemma 31 (which would give $\alpha'_i = 0$ or $\beta^*_i = 0$) to show the general case

(b) Here we show that ALG can pack further items aligned with the bottom of the strip before generating the gap $\beta^*_1 p^*_1$

Figure 32: Illustrations for Lemmas 35 and 36

By Equation (6.3) we have $(1 - \alpha^*_i)p^*_{i+1} = (\beta^*_i + 1)p^*_i$ and finally get

$$\frac{\gamma^*_{i+1} + \beta^*_{i+1} - (\rho - 2)\alpha^*_{i+1}}{1 - \alpha^*_{i+1}} = \frac{\gamma^*_i + (\rho - 1)\frac{q_{i+1}}{p^*_i} + (\rho - 1)\beta^*_i - 1}{1 + \beta^*_i}. \qquad \square$$

This completes our preparations for the induction that we show next.

THE INDUCTION Now we give the intended contradiction. On the one hand, any ρ-competitive algorithm needs to satisfy $\Phi_i \geq 0$. We show, in the contrary, that the potential Φ_i indefinitely decreases.

We start the induction with the next lemma, giving a maximal initial value of $\rho - 2 + (\rho - 1)\hat{c}$ for the potential. Afterwards, we distinguish three cases according to the definition of q_{i+1}. If $q_{i+1} = \beta^*_i p^*_i$ or $q_{i+1} = \alpha'_i p^*_i$ we show $\Phi_{i+1} \leq \Phi_i - \varepsilon$ for some $\varepsilon > 0$. The case $q_{i+1} = q_i$ is more involved. Either we also get a decreasing potential or the potential might actually rise, but is still lower than the initial value. Therefore, this rise can only happen once, as we finally show when we bring together all parts.

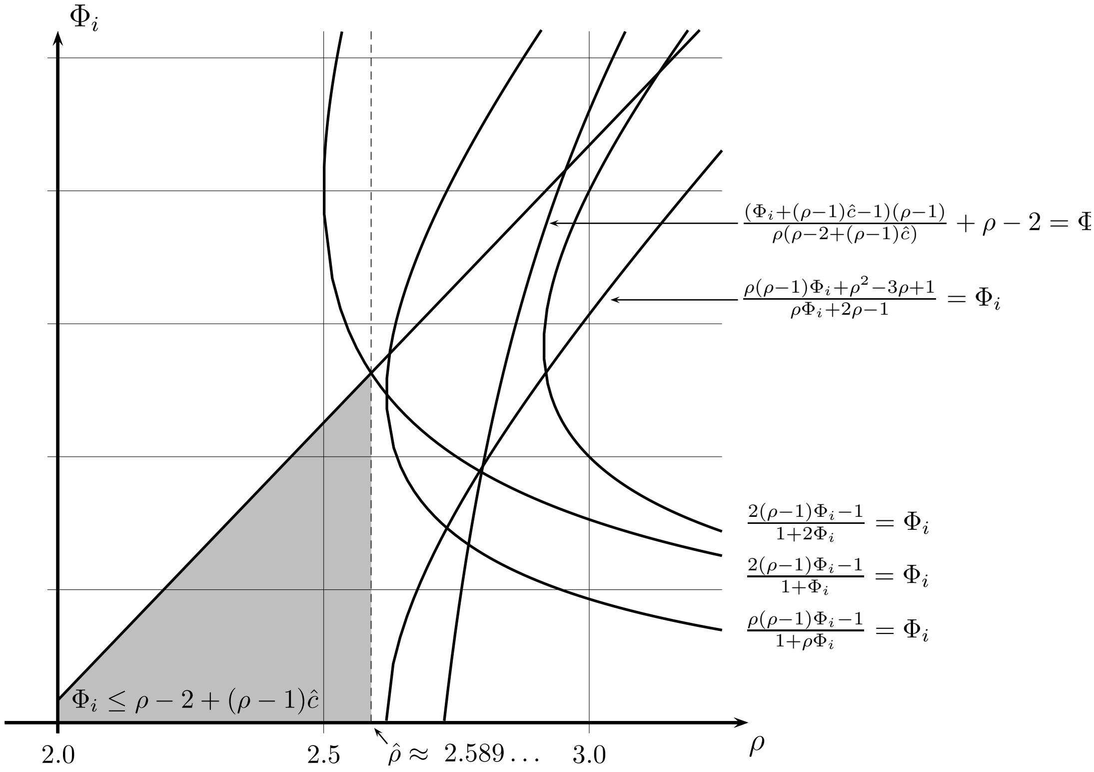


Figure 33: The initial upper bound for the potential (forcing the potential into the shaded area) along with the conditions. In particular, $\Phi_i \leq \rho - 2 + (\rho-1)\hat{c}$ implies that for any $\rho < \hat{\rho}$ we can find an $\varepsilon > 0$ such that $\Phi_{i+1} \leq \Phi_i - \varepsilon$

In the following calculations (which are partially very technical) we basically derive a series of upper bounds on the potential Φ_{i+1}. In detail, we get

$$\Phi_{i+1} \leq \frac{2(\rho-1)\Phi_i - 1}{1+\Phi_i} \qquad \text{in case } q_{i+1} = \beta_i^* p_i^* \tag{6.4}$$

$$\text{and} \qquad \Phi_{i+1} \leq \frac{\rho(\rho-1)\Phi_i - 1}{1+\rho\Phi_i} \qquad \text{in case } q_{i+1} = \beta_i^* p_i^*$$

$$\text{and} \qquad \Phi_{i+1} \leq \frac{2(\rho-1)\Phi_i - 1}{1+2\Phi_i} \qquad \text{in case } q_{i+1} = \alpha_i' p_i^*$$

$$\text{and} \qquad \Phi_{i+1} < \frac{\rho(\rho-1)\Phi_i + \rho^2 - 3\rho + 1}{\rho\Phi_i + 2\rho - 1} \qquad \text{in case } q_{i+1} = q_i$$

$$\text{and} \qquad \Phi_{i+1} \leq \frac{(\Phi_i + (\rho-1)\hat{c} - 1)(\rho-1)}{\rho(\rho - 2 + (\rho-1)\hat{c})} + \rho - 2 \qquad \text{in case } q_{i+1} = q_i.$$

All these conditions eventually imply $\Phi_{i+1} \leq \Phi_i - \varepsilon$ for some $\varepsilon > 0$ and $\rho < \hat{\rho}$. Just for Condition (6.4) we additionally require the induction hypothesis $\Phi_i \leq \rho - 2 + (\rho-1)\hat{c}$. This is actually exactly the condition that gives us the value of $\hat{c} = \frac{1-\sqrt{4\hat{\rho}^2-12\hat{\rho}+5}}{2(\hat{\rho}-1)}$. See Figure 33 for an illustration of the conditions above.

We now start with the induction hypothesis. Not only do we give an upper bound for the initial potential Φ_1, but also for the ratio q_1/p_1^*. This is needed later when we bring the different parts together.

Lemma 36. *We have*

$$\Phi_1 \leq \rho - 2 + (\rho - 1)\hat{c}$$

and

$$\frac{q_1}{p_1^*} < \frac{1}{\rho}.$$

Proof. Consider the packing of ALG and the optimal packing after p_1^* is released—see Figure 32b. Recall that $\mathrm{ALG}(r_k)$ is the packing height at the end of the starting phase and that $q_1 = \hat{c}\,\mathrm{ALG}(r_k)$. As p_1^* equals the height of the interval below q_1 we have

$$\mathrm{OPT}(p_1^*) = p_1^* + q_1 = p_1^* + \hat{c}\,\mathrm{ALG}(r_k) \qquad \text{and}$$
$$\mathrm{ALG}(p_1^*) = p_1^* + q_1 + \beta_1^* p_1^* + p_1^* = 2p_1^* + \hat{c}\,\mathrm{ALG}(r_k) + \beta_1^* p_1^*.$$

Moreover, we have $p_1^* = \mathrm{ALG}(r_k) + \alpha_1^* p_1^*$ and thus $p_1^* = \frac{\mathrm{ALG}(r_k)}{1-\alpha_1^*}$. We get

$$\begin{aligned}
\gamma_1^* p_1^* &= \rho\,\mathrm{OPT}(p_1^*) - \mathrm{ALG}(p_1^*) \\
&= (\rho - 2)p_1^* + (\rho - 1)\,\hat{c}\,\mathrm{ALG}(r_k) - \beta_1^* p_1^* \\
\Rightarrow \quad \Phi_i = \gamma_1^* + \beta_1^* &= \rho - 2 + (\rho - 1)\hat{c}(1 - \alpha_1^*) \qquad \text{since } p_1^* = \frac{\mathrm{ALG}(r_k)}{1 - \alpha_1^*} \\
&\leq \rho - 2 + (\rho - 1)\hat{c}.
\end{aligned}$$

Finally, observe that

$$\frac{q_1}{p_1^*} \leq \frac{\hat{c}\,\mathrm{ALG}(r_k)}{\mathrm{ALG}(r_k)} = \hat{c} < \frac{1}{\rho}. \qquad \square$$

With the next two lemmas we show that the potential decreases if $q_{i+1} = \beta_i^* p_i^*$ or $q_{i+1} = \alpha_i' p_i^*$. At the same time we show that q_{i+1}/p_{i+1}^* is bounded, which we need in the last case $q_{i+1} = q_i$.

Lemma 37. *If $\Phi_i \leq \rho - 2 + (\rho - 1)\hat{c}$ and $q_{i+1} = \beta_i^* p_i^*$, then*

$$\Phi_{i+1} \leq \Phi_i - \varepsilon \qquad \text{for some } \varepsilon > 0$$

and

$$\frac{q_{i+1}}{p_{i+1}^*} \leq \frac{\Phi_{i+1}}{\rho - 1} \quad \text{or} \quad \frac{q_{i+1}}{p_{i+1}^*} < \frac{1}{\rho}.$$

Proof. By Lemma 31 Condition 31.1 we can assume $\alpha_i' = 0$. Thus Lemma 35 yields

$$\Phi_{i+1} = \frac{\gamma_i^* + 2(\rho - 1)\beta_i^* - 1}{1 + \beta_i^*}(1 - \alpha_{i+1}^*) + (\rho - 2)\alpha_{i+1}^*.$$

Note that this function is linear in α_{i+1}^*. Thus Φ_{i+1} attains its maximum for maximal or minimal α_{i+1}^*. We show for both cases that $\Phi_{i+1} \leq \Phi_i - \varepsilon$ for some $\varepsilon > 0$.

If Φ_{i+1} is non-increasing in α^*_{i+1} we have

$$\begin{aligned}
\Phi_{i+1} &\leq \frac{\gamma^*_i + 2(\rho-1)\beta^*_i - 1}{1+\beta^*_i} && \text{as } \alpha^*_{i+1} \geq 0 \\
&= \frac{\Phi_i + (2\rho-3)\beta^*_i - 1}{1+\beta^*_i} && \text{as } \gamma^*_i + \beta^*_i = \Phi_i \\
&\leq \frac{2(\rho-1)\Phi_i - 1}{1+\Phi_i}.
\end{aligned}$$

The last step holds as $\beta^*_i \leq \Phi_i$ and the function is increasing with respect to β^*_i. With $\varepsilon = \varepsilon(\rho) = \frac{\hat{c}^2(\rho-1)^2 - \hat{c}(\rho-1) - (\rho-1)(\rho-2)+1}{(\rho-1)(1+\hat{c})}$ we have $\Phi_{i+1} \leq \Phi_i - \varepsilon$ since

$$\begin{aligned}
& \frac{2(\rho-1)\Phi_i - 1}{1+\Phi_i} \leq \Phi_i - \varepsilon \\
\Leftrightarrow \quad & \Phi_i^2 - (2\rho - 3 + \varepsilon)\Phi_i \geq \varepsilon - 1 \\
\Leftarrow \quad & \big(\rho - 2 + (\rho-1)\hat{c}\big)^2 - (2\rho - 3 + \varepsilon)(\rho - 2 + (\rho-1)\hat{c}) \geq \varepsilon - 1 \\
& \text{as } \Phi_i \leq \rho - 2 + (\rho-1)\hat{c} \text{ and } 2\Phi_i - 2\rho + 3 - \varepsilon \leq 2(\rho-1)\hat{c} - 1 < 0 \\
\Leftrightarrow \quad & \frac{\big(\rho - 2 + (\rho-1)\hat{c}\big)^2 - (2\rho-3)(\rho - 2 + (\rho-1)\hat{c}) + 1}{1 + \rho - 2 + (\rho-1)\hat{c}} \geq \varepsilon \\
\Leftarrow \quad & \frac{\hat{c}^2(\rho-1)^2 - \hat{c}(\rho-1) - (\rho-1)(\rho-2) + 1}{(\rho-1)(1+\hat{c})} = \varepsilon
\end{aligned}$$

It remains to show $\varepsilon = \varepsilon(\rho) > 0$. We have $\varepsilon(\hat{\rho}) = 0$ since

$$\hat{c}^2(\hat{\rho}-1)^2 - \hat{c}(\hat{\rho}-1) - (\hat{\rho}-1)(\hat{\rho}-2) + 1 = 0$$

for $\hat{c} = \frac{1-\sqrt{4\hat{\rho}^2 - 12\hat{\rho} + 5}}{2(\hat{\rho}-1)}$. Now observe that ε is strictly decreasing with ρ since

$$\frac{\partial}{\partial\rho}\big(\varepsilon(\rho)\big) = \frac{(\hat{c}^2-1)(\rho-1)^2 - (\rho^2 - 2\rho + 2)}{(\rho-1)^2(1+\hat{c})} < 0$$

as $\hat{c}^2 - 1 < 0$ and $\rho^2 - 2\rho + 2 > 0$. Thus we have $\varepsilon = \varepsilon(\rho) > \varepsilon(\hat{\rho}) = 0$ in this case.

Now, if Φ_{i+1} is increasing in α^*_{i+1}, we use Lemma 33 to get

$$\begin{aligned}
\Phi_{i+1} &\leq \frac{\gamma^*_i + 2(\rho-1)\beta^*_i - 1}{1+\beta^*_i} \cdot \Big(1 - \frac{\gamma^*_i + (\rho-1)\beta^*_i}{1+\gamma^*_i + \rho\beta^*_i}\Big) + (\rho-2)\cdot\frac{\gamma^*_i + (\rho-1)\beta^*_i}{1+\gamma^*_i+\rho\beta^*_i} \\
&\leq \frac{\gamma^*_i + 2(\rho-1)\beta^*_i - 1}{1+\beta^*_i} \cdot \frac{1+\beta^*_i}{1+\gamma^*_i+\rho\beta^*_i} + (\rho-2)\cdot\frac{\gamma^*_i + (\rho-1)\beta^*_i}{1+\gamma^*_i+\rho\beta^*_i} \\
&= \frac{(\rho-1)\gamma^*_i + \rho(\rho-1)\beta^*_i - 1}{1+\gamma^*_i+\rho\beta^*_i} \\
&= \frac{(\rho-1)\Phi_i + (\rho-1)^2\beta^*_i - 1}{1+\Phi_i + (\rho-1)\beta^*_i} && \text{as } \gamma^*_i + \beta^*_i = \Phi_i \\
&\leq \frac{\rho(\rho-1)\Phi_i - 1}{1+\rho\Phi_i}.
\end{aligned}$$

Again, the last step holds as $\beta_i^* \le \Phi_i$ and the function is increasing with respect to β_i^*. With $\varepsilon = 3 - \rho - 1/\rho > 0$ (for $\rho < \hat{\rho}$) we have $\Phi_{i+1} \le \Phi_i - \varepsilon$ since

$$\begin{aligned}
& & \frac{\rho(\rho-1)\Phi_i - 1}{1+\rho\Phi_i} &\le \Phi_i - 3 + \rho + \frac{1}{\rho} \\
&\Leftrightarrow & \rho(\rho-1)\Phi_i - 1 &\le \Phi_i - 3 + \rho + \frac{1}{\rho} + \rho\Phi_i^2 - 3\rho\Phi_i + \rho^2\Phi_i + \Phi_i \\
&\Leftrightarrow & \Phi_i^2 + \Big(\frac{2-2\rho}{\rho}\Big)\Phi_i &\ge \frac{2-\rho-\frac{1}{\rho}}{\rho} \\
&\Leftrightarrow & \Big(\Phi_i^2 + \frac{1-\rho}{\rho}\Big)^2 &\ge \Big(\frac{1-\rho}{\rho}\Big)^2 + \frac{2-\rho-\frac{1}{\rho}}{\rho} = 0
\end{aligned}$$

Thus in both cases we have $\Phi_{i+1} \le \Phi_i - \varepsilon$ for some $\varepsilon > 0$.

It remains to show

$$\frac{q_{i+1}}{p_{i+1}^*} \le \frac{\Phi_{i+1}}{\rho-1} \qquad \text{or} \qquad \frac{q_{i+1}}{p_{i+1}^*} < \frac{1}{\rho}.$$

We have $\frac{q_{i+1}}{p_{i+1}^*} = \frac{\beta_i^*}{1+\beta_i^*}(1-\alpha_{i+1}^*)$ by Equation (6.3). If $\beta_i^* < \frac{1}{\rho-1}$ we have

$$\frac{q_{i+1}}{p_{i+1}^*} = \frac{\beta_i^*}{1+\beta_i^*}(1-\alpha_{i+1}^*) \le \frac{\beta_i^*}{1+\beta_i^*} < \frac{1}{\rho-1} \cdot \frac{1}{1+\frac{1}{\rho-1}} = \frac{1}{\rho-1} \cdot \frac{1}{\frac{\rho}{\rho-1}} = \frac{1}{\rho}.$$

Otherwise, we have $\beta_i^* \ge \frac{1}{\rho-1}$ and thus

$$\begin{aligned}
& & \frac{q_{i+1}}{p_{i+1}^*} = \frac{\beta_i^*}{1+\beta_i^*}(1-\alpha_{i+1}^*) &\le \frac{\Phi_{i+1}}{\rho-1} \\
&\Leftrightarrow & (\rho-1)\beta_i^* &\le \gamma_i^* + 2(\rho-1)\beta_i^* - 1 + \underbrace{(\rho-2)(1+\beta_i^*)\frac{\alpha_{i+1}^*}{1-\alpha_{i+1}^*}}_{\ge 0} \\
&\Leftarrow & 1 &\le \gamma_i^* + (\rho-1)\beta_i^* \\
&\Leftarrow & \beta_i^* &\ge \frac{1}{\rho-1}.
\end{aligned}$$

□

Lemma 38. *If $\Phi_i \le \rho - 2 + (\rho-1)\hat{c}$ and $q_{i+1} = \alpha_i' p_i^*$ then*

$$\begin{aligned}
& \Phi_{i+1} \le \Phi_i - \varepsilon && \textit{for some } \varepsilon > 0 \\
\textit{and} \quad & \frac{q_{i+1}}{p_{i+1}^*} < \frac{1}{\rho}.
\end{aligned}$$

Proof. In this case we can assume $\beta_i^* = 0$ (by Lemma 31 Condition 31.2) and hereby have $\Phi_i = \gamma_i^*$. Thus by Lemma 35 and with $(\rho-1)\alpha_i' \le \gamma_i^* = \Phi_i$ by Lemma 34 we have

$$\begin{aligned}
\Phi_{i+1} &= (\Phi_i + (\rho-1)\alpha_i' - 1)(1-\alpha_{i+1}^*) + (\rho-2)\alpha_{i+1}^* \\
&\le (2\Phi_i - 1)(1-\alpha_{i+1}^*) + (\rho-2)\alpha_{i+1}^*.
\end{aligned}$$

We consider the derivative with respect to α^*_{i+1} and with $\Phi_i \le \rho - 2 + (\rho-1)\hat{c}$ we get

$$\frac{\partial}{\partial \alpha^*_{i+1}}\Big((2\Phi_i - 1)(1-\alpha^*_{i+1}) + (\rho-2)\alpha^*_{i+1}\Big) = \rho - 1 - 2\Phi_i \ge 3 - \rho - 2(\rho-1)\hat{c} > 0.$$

Thus Φ_{i+1} increases with α^*_{i+1} and since by Lemma 33

$$\alpha^*_{i+1} \le \frac{\gamma^*_i + (\rho-1)\alpha'_i}{1+\gamma^*_i + (\rho-1)\alpha'_i} \le \frac{2\gamma^*_i}{1+2\gamma^*_i} = \frac{2\Phi_i}{1+2\Phi_i}$$

we get

$$\begin{aligned}\Phi_{i+1} &\le (2\Phi_i - 1)\Big(1 - \frac{2\Phi_i}{1+2\Phi_i}\Big) + (\rho-2)\frac{2\Phi_i}{1+2\Phi_i} \\ &= \frac{2(\rho-1)\Phi_i - 1}{1+2\Phi_i}.\end{aligned}$$

With $\varepsilon = \sqrt{2\rho} - \rho + 1/2 > 0$ (for $\rho < \hat{\rho}$) we have $\Phi_{i+1} \le \Phi_i - \varepsilon$ since

$$\begin{aligned} && \frac{2(\rho-1)\Phi_i - 1}{1+2\Phi_i} &\le \Phi_i - \sqrt{2\rho} + \rho - \frac{1}{2} \\ \Leftrightarrow && 2\Phi_i^2 + (2 - 2\sqrt{2\rho})\Phi_i &\ge \sqrt{2\rho} - \rho - \frac{1}{2} \\ \Leftrightarrow && \left(\Phi_i + \frac{1-\sqrt{2\rho}}{2}\right)^2 &\ge \left(\frac{1-\sqrt{2\rho}}{2}\right)^2 + \frac{\sqrt{2\rho} - \rho - \frac{1}{2}}{2} = 0. \end{aligned}$$

Thus we proved the first part of the lemma.

It remains to show

$$\frac{q_{i+1}}{p^*_{i+1}} < \frac{1}{\rho}.$$

With $\beta^*_i = 0$ we have $p^*_{i+1} = p^*_i/(1-\alpha^*_{i+1})$ by Equation (6.3). Using $\alpha^*_{i+1} \ge \frac{(\rho-1)\alpha'_i}{1+(\rho-1)\alpha'_i}$ by Lemma 32 and $\alpha'_i \le \alpha^*_i < 1$ (by definition of α^*_i as a fraction of p^*_i) we get

$$\frac{q_{i+1}}{p^*_{i+1}} = \frac{\alpha'_i p^*_i}{p^*_{i+1}} = \alpha'_i(1-\alpha^*_{i+1}) \le \frac{\alpha'_i}{1+(\rho-1)\alpha'_i} < \frac{1}{\rho}.$$

This finishes the proof of this lemma. □

Finally, we consider the case $q_{i+1} = q_i$. First, we show that the potential definitely decreases if $q_i/p^*_i < 1/\rho$.

Lemma 39. *If $\Phi_i \le \rho - 2 + (\rho-1)\hat{c}$ and $\frac{q_i}{p^*_i} < \frac{1}{\rho}$ and $q_{i+1} = q_i$ then*

$$\Phi_{i+1} \le \Phi_i - \varepsilon \qquad \textit{for some } \varepsilon > 0$$

and

$$\frac{q_{i+1}}{p^*_{i+1}} < \frac{1}{\rho}.$$

Proof. The second part is trivial since $q_{i+1} = q_i$, $p^*_{i+1} \geq p^*_i$ and $q_i/p^*_i < 1/\rho$.

To show the first part we assume $\alpha'_i = 0$ and $\beta^*_i = 0$ according to Conditions 31.1 and 31.2 of Lemma 31. Thus we have $\Phi_i = \gamma^*_i$ and with Lemma 35 we get

$$\Phi_{i+1} = \big(\Phi_i + (\rho-1)\frac{q_i}{p^*_i} - 1\big)(1-\alpha^*_{i+1}) + (\rho-2)\alpha^*_{i+1}$$

We consider the derivative with respect to α^*_{i+1} and get

$$\frac{\partial}{\partial\alpha^*_{i+1}}\big(\Phi_{i+1}\big) = \rho - 1 - \Phi_i - (\rho-1)\frac{q_i}{p^*_i} > 1 - (\rho-1)\hat{c} - \frac{\rho-1}{\rho} > 0.$$

Thus Φ_{i+1} increases with α^*_{i+1} and since by Lemma 33

$$\alpha^*_{i+1} \leq \frac{\Phi_i + (\rho-1)\frac{q_i}{p^*_i}}{1 + \Phi_i + (\rho-1)\frac{q_i}{p^*_i}}$$

we get

$$\begin{aligned}
\Phi_{i+1} &\leq \frac{(\rho-1)\Phi_i + (\rho-1)^2\frac{q_i}{p^*_i} - 1}{1 + \Phi_i + (\rho-1)\frac{q_i}{p^*_i}} \\
&< \frac{(\rho-1)\Phi_i + \frac{(\rho-1)^2}{\rho} - 1}{\Phi_i + \frac{2\rho-1}{\rho}} && \text{as } \frac{q_i}{p^*_i} < \frac{1}{\rho} \\
&= \frac{\rho(\rho-1)\Phi_i + \rho^2 - 3\rho + 1}{\rho\Phi_i + 2\rho - 1}.
\end{aligned}$$

With

$$\varepsilon = \frac{3\rho - \rho^2 - 1}{\rho\Phi_i + 2\rho - 1} \geq \frac{3\rho - \rho^2 - 1}{\rho(\rho + (\rho-1)\hat{c}) - 1} > 0,$$

as $\Phi_i \leq \rho - 2 + (\rho-1)\hat{c}$ and $3\rho - \rho^2 - 1 > 0$ for $\rho < \hat{\rho}$, we have $\Phi_{i+1} \leq \Phi_i - \varepsilon$ since

$$\begin{aligned}
& & \frac{\rho(\rho-1)\Phi_i + \rho^2 - 3\rho + 1}{\rho\Phi_i + 2\rho - 1} &\leq \Phi_i - \frac{3\rho - \rho^2 - 1}{\rho\Phi_i + 2\rho - 1} \\
&\Leftrightarrow & \rho(\rho-1)\Phi_i + \rho^2 - 3\rho + 1 &\leq \rho\,\Phi_i^2 + (2\rho-1)\Phi_i - 3\rho + \rho^2 + 1 \\
&\Leftrightarrow & 0 &\leq \rho\Phi_i^2 + (3\rho - \rho^2 - 1)\Phi_i
\end{aligned}$$

which is satisfied as $3\rho - \rho^2 - 1 > 0$ for $\rho < \hat{\rho}$. □

If we do not have $q_i/p^*_i < 1/\rho$ we can still assume $q_i/p^*_i \leq (\rho - 2 + (\rho-1)\hat{c})/(\rho-1)$ by Lemmas 37 and 38 (as $\Phi_i \leq \rho - 2 + (\rho-1)\hat{c}$ and this ratio does not increase in case $q_{i+1} = q_i$).

We use this bound to show that either the potential still decreases or we can bound the potential by $\rho - 2$ and the q_{i+1}/p^*_{i+1} ratio is less than $1/\rho$. So from this point on we remain in the case of the previous lemma and the potential decreases by a constant in every step.

Lemma 40. *If* $\Phi_i \le \rho - 2 + (\rho-1)\hat{c}$ *and* $\frac{q_i}{p_i^*} \le \frac{\rho-2+(\rho-1)\hat{c}}{\rho-1}$ *and* $q_{i+1} = q_i$ *then either*

$$\Phi_{i+1} < \rho - 2 \qquad \text{and}$$
$$\frac{q_{i+1}}{p_{i+1}^*} < \frac{1}{\rho},$$

or

$$\Phi_{i+1} \le \Phi_i - \varepsilon \qquad \text{for some } \varepsilon > 0\text{, and}$$
$$\frac{q_{i+1}}{p_{i+1}^*} \le \frac{\rho-2+(\rho-1)\hat{c}}{\rho-1}.$$

Proof. As in Lemma 39 we have $\alpha_i' = 0$, $\beta_i^* = 0$, $\Phi_i = \gamma_i^*$ and

$$\Phi_{i+1} = \big(\Phi_i + (\rho-1)\frac{q_i}{p_i^*} - 1\big)(1-\alpha_{i+1}^*) + (\rho-2)\alpha_{i+1}^*$$

Again, Φ_{i+1} increases with α_{i+1}^*. We distinguish two cases according to the value of α_{i+1}^*.

If $\alpha_{i+1}^* > 1 - \frac{\rho-1}{\rho(\rho-2+(\rho-1)\hat{c})}$ then

$$\frac{q_{i+1}}{p_{i+1}^*} = \frac{q_i}{p_i^*}(1-\alpha_{i+1}^*) < \frac{\rho-2+(\rho-1)\hat{c}}{\rho-1} \cdot \frac{\rho-1}{\rho(\rho-2+(\rho-1)\hat{c})} = \frac{1}{\rho}.$$

As Φ_{i+1} is increasing with α_{i+1}^* we use Lemma 33 to get

$$\begin{aligned}\Phi_{i+1} &\le \frac{(\rho-1)\Phi_i + (\rho-1)^2\frac{q_i}{p_i^*} - 1}{1 + \Phi_i + (\rho-1)\frac{q_i}{p_i^*}} \\ &\le \frac{2(\rho-1)(\rho-2+(\rho-1)\hat{c}) - 1}{1 + 2(\rho-2+(\rho-1)\hat{c})}\end{aligned}$$

as $\Phi_i \le \rho-2+(\rho-1)\hat{c}$ and $\frac{q_i}{p_i^*} \le \frac{\rho-2+(\rho-1)\hat{c}}{\rho-1}$. We have

$$\begin{aligned} & \frac{2(\rho-1)(\rho-2+(\rho-1)\hat{c}) - 1}{1 + 2(\rho-2+(\rho-1)\hat{c})} < \rho - 2 \\ \Leftrightarrow \quad & 2(\rho-1)(\rho-2+(\rho-1)\hat{c}) - 1 < \rho - 2 + 2(\rho-2)(\rho-2+(\rho-1)\hat{c}) \\ \Leftrightarrow \quad & 2(\rho-2+(\rho-1)\hat{c}) < \rho - 1 \\ \Leftrightarrow \quad & 2(\rho-1)\hat{c} < 3 - \rho, \end{aligned}$$

which holds for $\rho < \hat{\rho}$. We showed that if $\alpha_{i+1}^* > 1 - \frac{\rho-1}{\rho(\rho-2+(\rho-1)\hat{c})}$, then we have $\Phi_{i+1} < \rho - 2$ and $q_{i+1}/p_{i+1}^* < 1/\rho$.

Otherwise, we have $\alpha_{i+1}^* \le 1 - \frac{\rho-1}{\rho(\rho-2+(\rho-1)\hat{c})}$ and get

$$\begin{aligned}\Phi_{i+1} &\le \frac{(\Phi_i + (\rho-1)\frac{q_i}{p_i^*} - 1)(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} + \rho - 2 - \frac{(\rho-2)(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} \\ &\le \frac{(\Phi_i + (\rho-1)\hat{c} - 1)(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} + \rho - 2\end{aligned}$$

as $\frac{q_i}{p_i^*} \leq \frac{\rho-2+(\rho-1)\hat{c}}{\rho-1}$. With $\varepsilon = \Phi_i + \frac{(1-(\rho-1)\hat{c}-\Phi_i)(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} - \rho + 2$ we have $\Phi_{i+1} \leq \Phi_i - \varepsilon$ since

$$\frac{(\Phi_i + (\rho-1)\hat{c} - 1)(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} + \rho - 2 \leq \Phi_i - \Phi_i - \frac{(1-(\rho-1)\hat{c}-\Phi_i)(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} + \rho - 2.$$

It remains to show $\varepsilon > 0$. To see this, observe that ε is increasing with respect to Φ_i as

$$\frac{\partial}{\partial \Phi_i}\Big(\Phi_i + \frac{(1-(\rho-1)\hat{c}-\Phi_i)(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} - \rho + 2\Big) = 1 - \frac{\rho-1}{\rho(\rho-2+(\rho-1)\hat{c})} > 0$$

for $2.55 \leq \rho < \hat{\rho}$ (here we use that we assume $\delta = \hat{\rho} - \rho$ is sufficiently small). As $\Phi_i \geq 0$ we have

$$\varepsilon \geq \frac{(1-(\rho-1)\hat{c})(\rho-1)}{\rho(\rho-2+(\rho-1)\hat{c})} - \rho + 2 > 0.$$

Of course,

$$\frac{q_{i+1}}{p_{i+1}^*} \leq \frac{q_i}{p_i^*} \leq \frac{\rho-2+(\rho-1)\hat{c}}{\rho-1}$$

holds trivially. We showed that if $\alpha_{i+1}^* \leq 1 - \frac{\rho-1}{\rho(\rho-2+(\rho-1)\hat{c})}$, then we have $\Phi_{i+1} \leq \Phi_i - \varepsilon$ for $\varepsilon > 0$ and $q_{i+1}/p_{i+1}^* \leq (\rho-2+(\rho-1)\hat{c})/(\rho-1)$. This finishes the proof of this lemma. □

This ends this extensive induction. Let us summarize the complete induction and show that it actually gives the desired contradiction.

Recall that our induction hypothesis in Lemma 36 states that $\Phi_1 \leq \rho - 2 + (\rho-1)\hat{c}$ and $q_1/p_1^* < 1/\rho$.

First assume that whenever we need to apply Lemma 40, then the second outcome is valid, i.e., $\Phi_{i+1} \leq \Phi_i - \varepsilon$ and $q_{i+1}/p_{i+1}^* \leq (\rho-2+(\rho-1)\hat{c})/(\rho-1)$. Then Lemma 37 (for $q_{i+1} = \beta_i^* p_i^*$), Lemma 38 (for $q_{i+1} = \alpha_i' p_i^*$) and Lemmas 39 and 40 (for $q_{i+1} = q_i$) show that the potential decreases by a constant in every step.

Now if Lemma 40 is applied and the second outcome is invalid, then we afterwards have $\Phi_{i+1} < \rho - 2$ and $q_{i+1}/p_{i+1}^* < 1/\rho$. Thus Lemma 37 (for $q_{i+1} = \beta_i^* p_i^*$), Lemma 38 (for $q_{i+1} = \alpha_i' p_i^*$) and Lemmas 39 (for $q_{i+1} = q_i$) show that the q_i/p_i^* ratio remains less than $1/\rho$. So the precondition for Lemma 39 is always satisfied if $q_{i+1} = q_i$ and we do not need to apply Lemma 40 anymore. Thus from this point on, the potential decreases by a constant in every further step.

In total we showed a contradiction to $\Phi_i \geq 0$ for all $i \geq 1$ and thus proved Theorem 10.

6.3 UPPER BOUND

In this section we present the online algorithm ONL for packing primitive sequences that consist solely of thin and blocking items. We prove that the competitive ratio of

Algorithm 9 Online algorithm for packing primitive sequences

Initially the packing is considered to be blocked
whenever a rectangle r_j is released **do**
    if r_j is a blocking item **then**
        Pack r_j at the lowest possible height
    else if r_j is a thin item **then**
        if the packing is open **then**
            Pack r_j bottom-aligned with the top thin item
        else if the packing is blocked **then**
            Try to pack r_j below the top item
            If this is not possible, pack r_j at distance $(\rho - 2)r_j$ above the packing

ONL is $\rho = (3 + \sqrt{5})/2$. As we will see, our algorithm neglects the starting phase. In the final discussion in Section 6.4 we will see why this is the case.

We differentiate two kinds of packings according to the item on top: If the item on top of the packing is a blocking item, we have a *blocked* packing, otherwise we have an *open* packing. Initially, we have a blocked packing by considering the bottom of the strip as a blocking item of height 0.

The general idea of the algorithm ONL is pretty straight-forward: Generate a β-gap of relative height $\rho - 2$ whenever a jump is unavoidable and pack arriving blocking items as low as possible. Since we neglect the starting phase, $\beta = \rho - 2$ is the maximal β-gap that we can ensure. This leads to the following algorithm—see also Algorithm 9.

- If a blocking item r_j arrives, we pack r_j at the lowest possible height. This can be inside the packing, if a sufficiently large gap is available, or directly on top of the packing. In the latter case, the packing is blocked afterwards.
- If a thin item r_j arrives at an open packing, we bottom-align r_j with the top item.
- If, finally, a thin item r_j arrives at a blocked packing, we try to pack r_j below the blocking item on top. If this is not possible, i.e., r_j exceeds the height of all intervals for thin items, we pack r_j at distance $\beta r_j = (\rho - 2)r_j$ above the top of the packing. This changes the packing to an open packing again.

We show that ONL is ρ-competitive for $\rho = (3 + \sqrt{5})/2$. Actually, this is only questionable in one case, namely, when we pack a thin item r_j with distance $(\rho - 2)r_j$ above the packing. All other cases are trivial since if the packing height increases, then the optimal height increases by the same value (for thin items the packing height only increases if r_j is the new maximal item).

We denote the thin items that are packed when generating a new gap by s_i for the i-th jump. Let s'_{i-1} be the highest thin item that is bottom-aligned with s_{i-1}. Note that the blocking item that blocks the packing after the i-th jump is packed directly above s'_{i-1}. See Figure 34 for an illustration.

It is obvious that the first jump item s_1, that is actually the first thin item that arrives, can be packed.

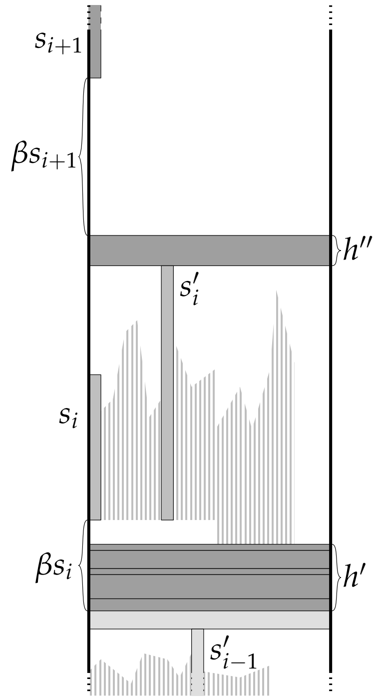


Figure 34: Packing after the $(i+1)$-th jump. The blocking items that arrived after s_i are shown in darker shade. By definition, s_i is the first item that does not fit into the previous interval. Thus we have $s_{i+1} > s'_i + \beta s_i - h'$

For the induction step we assume $\mathrm{ONL}(s_i) \leq \rho\,\mathrm{OPT}(s_i)$. Before a jump can become unavoidable, new blocking items of total height greater than βs_i need to arrive as otherwise the gap below s_i could accommodate all of them. Let h' be the height of the blocking items that are packed into the β-gap below s_i and let h'' be the total height of blocking items that arrive between s_i and s_{i+1} and are packed above s_i. We have $h' \leq (\rho-2)s_i$ and $h' + h'' > (\rho-2)s_i$ as otherwise no blocking item would be packed on top. As further blocking items could be packed even below s'_{i-1} we get

$$\mathrm{OPT}(s_{i+1}) \geq \mathrm{OPT}(s_i) + h' + h'' + s_{i+1} - s_i$$
$$\mathrm{ONL}(s_{i+1}) = \mathrm{ONL}(s_i) + s'_i - s_i + h'' + \beta s_{i+1} + s_{i+1}.$$

And thus we have

$$\begin{aligned} & \mathrm{ONL}(s_{i+1}) \leq \rho\,\mathrm{OPT}(s_{i+1}) \\ \Leftarrow\quad & \mathrm{ONL}(s_i) + s'_i - s_i + h'' + \beta s_{i+1} + s_{i+1} \leq \rho\big(\mathrm{OPT}(s_i) + h' + h'' + s_{i+1} - s_i\big) \\ \Leftarrow\quad & (\rho-1)s_i + s'_i - \rho h' - (\rho-1)h'' \leq (\rho-1-\beta)s_{i+1}. \end{aligned}$$

As $\rho - 1 - \beta = 1$ and $s_{i+1} > s'_i + (\rho-2)s_i - h'$ this is satisfied if

$$\begin{aligned} & (\rho-1)s_i + s'_i - \rho h' - (\rho-1)h'' \leq s'_i + (\rho-2)s_i - h' \\ \Leftrightarrow\quad & s_i \leq (\rho-1)(h' + h'') \\ \Leftarrow\quad & s_i \leq (\rho-1)(\rho-2)s_i = s_i. \end{aligned}$$

The last equality holds since $\rho = (3+\sqrt{5})/2$ and thus $(\rho-1)(\rho-2) = 1$. Summarizing we showed the following theorem.

Theorem 11. ONL *is a ρ-competitive algorithm for packing primitive sequences with*

$$\rho = \frac{3+\sqrt{5}}{2} \approx 2.618.$$

6.4 FINAL DISCUSSION

In the previous sections we gave a lower bound of $\rho \geq 2.589\ldots$ and an upper bound of $\rho \leq 2.618\ldots$ for the competitive ratio ρ of the strip packing problem restricted to primitive sequences consisting solely of thin and blocking items. In this section we recapitulate the significance of the starting phase for the lower bound. Moreover, we present an approach to improve the lower bound and describe the challenges to improve our upper bound.

We analyzed the starting phase to derive Lemma 36, the starting point of our induction, where we showed $\Phi_1 \leq \rho - 2 + (\rho - 1)\hat{c}$. Observe that from all the conditions we derived in Section 6.2.2, only Condition (6.4), which was

$$\frac{2(\rho-1)\Phi_i - 1}{1+\Phi_i} \leq \Phi_i - \varepsilon,$$

actually required $\Phi_i \leq \rho - 2 + (\rho-1)\hat{c}$. It is easy to see (c.f. Figure 33) that all other conditions hold for $\rho < (3+\sqrt{5})/2$ regardless of the initial potential. Thus by improving the bound for Φ_1 in Lemma 36 we would directly get a better overall lower bound on the competitive ratio. Straightforward calculation yields that for $\Phi_i \leq \rho - 2 + x$ (for $x \geq 0$) we get a lower bound on the competitive ratio of

$$\hat{\rho} = \frac{3+\sqrt{5+4x^2-4x}}{2}.$$

To see this, observe that for $\varepsilon(x) = \frac{x^2-x-(\rho-1)(\rho-2)+1}{\rho-1+x}$, Condition (6.4) is satisfied. And for $\rho < \hat{\rho} = \frac{3+\sqrt{5+4x^2-4x}}{2}$ we have $\varepsilon(x) > 0$. Thus for $x = 0$ we even get the best possible lower bound of $(3+\sqrt{5})/2$.

Therefore, the most promising approach for improving the lower bound is to consider the starting phase and derive a better bound for Φ_1. In Lemma 36 we use $p_1^* = \mathrm{ALG}(r_k)/(1-\alpha_1^*)$ to show Condition (6.4). This holds since we do not force the algorithm to pack blocking items into the gaps that were kept open.

In the following we present a new objective for the starting phase that exploits this approach and that should eventually lead to an improved lower bound. Let $\lambda_{i-1}s_{i-1}$ and $\lambda_i s_i$ be the heights of the two last gaps. Now assume that no ρ-competitive algorithm can retain

$$\frac{\lambda_{i-1}s_{i-1}}{\mathrm{ALG}(r_k)} \geq \hat{d}$$

for some constant $\hat{d} > 0$ throughout the starting phase. In contrast to the starting phase that we analyzed in Section 6.2.1, we do not consider the current (maximal) gap $\lambda_i s_i$ here but rather the last gap before, i.e., $\lambda_{i-1}s_{i-1}$. As soon as the above "second-largest-gap-to-height" ratio drops below $\hat{d}$, we release a blocking item $q_1 = \hat{d}\,\mathrm{ALG}(r_k)$.

This blocking item can either be packed on top, leading to $\Phi_1 \leq \rho - 2 + (\rho - 1)\hat{d}$ as by the analyses in Lemma 36, or it can be packed into the gap of size $\lambda_i s_i$. In the latter case, we release a further blocking item q'_1 larger than $\lambda_i s_i - q_1$ which has to be packed on top. We have $\text{OPT}(p^*_1) = p^*_1 + q_1 + q'_1$ and $\text{ALG}(p^*_1) = \text{ALG}(s'_{i-1}) + q_1 + p^*_1 + q'_1 + \beta^*_1 p^*_1 + p^*_1$. And thus

$$\Phi_1 = \beta^*_1 + \gamma^*_1 = (\rho - 2) + \frac{(\rho - 1)(q_1 + q'_1) - \text{ALG}(s'_{i-1})}{p^*_1} \leq \rho - 2$$

in this case. The last step holds since $\text{ALG}(s'_{i-1}) \geq s_i \geq (\rho - 1)(q_1 + q'_1)$ (as $\text{ALG}(s_i) = s_i + \lambda_i s_i + \text{ALG}(s'_{i-1})$ and thus $q_1 + q_2 = \lambda_i s_i \leq \rho s_i - s_i - \text{ALG}(s'_{i-1}) \leq (\rho - 2)s_i$).

It is apparent that for the new objective we get $\hat{d} < \hat{c}$ (as we consider the "second-largest-gap-to-height" ratio instead of the max-gap-to-height ratio) and thus an improved lower bound on the competitive ratio. We believe that this idea can be further exploited by considering the whole series of gaps and releasing appropriate short blocking items. Unfortunately, analyzing these improved lower bounds requires parameters different from the well understood max-gap-to-height ratio and seems to be very involved.

One of the challenges to improve the upper bound relates to the approach described above. It is possible to retain a max-gap-to-height ratio of $\hat{c}$ with an $\hat{\rho}$-competitive algorithm when only thin items are released (generate gaps of maximal height whenever the max-gap-to-height ratio would otherwise be violated). But this does not lead to $\Phi_1 \geq \hat{\rho} - 2 + (\hat{\rho} - 1)\hat{c}$ as we cannot ensure $p^*_1 = \text{ALG}(r_k)/(1 - \alpha^*_1)$. The reason for this is that arriving blocking items need to be packed into the gaps to avoid blocking the packing with a short blocking item. But this leads to a potentially reduced interval height for the thin items and thus to a worse initial potential. An approach to overcome this difficulty might be to pack arriving blocking items into the second largest gap instead of the current gap. This would still give a lower bound on the overall height of blocking items received before blocking the packing is unavoidable. And still, p^*_1 can be large. Again, an analysis of this approach seems to be complicated.

Finally, the equation of Lemma 35, namely

$$\Phi_{i+1} = \frac{\gamma^*_i + (\rho - 1)\frac{q_{i+1}}{p^*_i} + (\rho - 1)\beta^*_i - 1}{1 + \beta^*_i}(1 - \alpha^*_{i+1}) + (\rho - 2)\alpha^*_{i+1}.$$

reveals another challenge to improve the upper bound. An adversary could, instead of releasing a blocking item q_{i+1} directly after the height of the thin item p^*_i exceeds the height of the previous interval, postpone the blocking item and release further thin items. This would increase the value of α^*_{i+1} if the thin items are packed bottom-aligned with p^*_i (as $\alpha^*_{i+1}p^*_{i+1}$ increases and this makes up an increasingly large fraction of p^*_{i+1}). As Lemma 35 shows, Φ_{i+1} tends to $\rho - 2$ as α^*_{i+1} approaches 1. Thus to sustain a potential larger than $\rho - 2$, the online algorithm would have to generate a new gap when receiving indefinitely growing thin items.

BIBLIOGRAPHY

[1] B. S. Baker, D. J. Brown, and H. P. Katseff. A 5/4 algorithm for two-dimensional packing. *Journal of Algorithms*, 2(4):348–368, 1981.

[2] B. S. Baker, E. G. Coffman, Jr., and R. L. Rivest. Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9(4):846–855, 1980.

[3] B. S. Baker and J. S. Schwarz. Shelf algorithms for two-dimensional packing problems. *SIAM Journal on Computing*, 12(3):508–525, 1983.

[4] N. Bansal. Personal communication, 2008.

[5] N. Bansal, A. Caprara, K. Jansen, L. Prädel, and M. Sviridenko. A structural lemma in 2-dimensional packing, and its implications on approximability. In *ISAAC: Proc. 20th International Symposium on Algorithms and Computation*, pages 77–86, 2009.

[6] N. Bansal, A. Caprara, and M. Sviridenko. A new approximation method for set covering problems, with applications to multidimensional bin packing. *SIAM Journal on Computing*, 39(4):1256–1278, 2009.

[7] N. Bansal, J. R. Correa, C. Kenyon, and M. Sviridenko. Bin packing in multiple dimensions - inapproximability results and approximation schemes. *Mathematics of Operations Research*, 31(1):31–49, 2006.

[8] D. J. Brown, B. S. Baker, and H. P. Katseff. Lower bounds for on-line two-dimensional packing algorithms. *Acta Informatica*, 18(2):207–225, 1982.

[9] A. Caprara. Packing d-dimensional bins in d stages. *Mathematics of Operations Research*, 33(1):203–215, 2008.

[10] A. Caprara, A. Lodi, and M. Monaci. Fast approximation schemes for two-stage, two-dimensional bin packing. *Mathematics of Operations Research*, 30(1):150–172, 2005.

[11] M. Chlebík and J. Chlebíková. Inapproximability results for orthogonal rectangle packing problems with rotations. In *CIAC: Proc. 6th Conference on Algorithms and Complexity*, pages 199–210, 2006.

[12] F. R. K. Chung, M. R. Garey, and D. S. Johnson. On packing two-dimensional bins. *SIAM Journal on Algebraic and Discrete Methods*, 3(1):66–76, 1982.

[13] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In *Approximation Algorithms for NP-hard Problems*, pages 46–93. PWS Publishing Company, 1997.

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

[15] D. Coppersmith and P. Raghavan. Multidimensional on-line bin packing: Algorithms and worst-case analysis. *Operations Research Letters*, 8(1):17–20, 1989.

[16] J. Csirik and G. J. Woeginger. Shelf algorithms for on-line strip packing. *Information Processing Letters*, 63(4):171–175, 1997.

[17] L. Epstein and R. van Stee. Online square and cube packing. *Acta Informatica*, 41(9):595–606, 2005.

[18] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $(1+\epsilon)$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[19] A. Freund and J. Naor. Approximating the advertisement placement problem. *Journal of Scheduling*, 7(5):365–374, 2004.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co, 1979.

[21] I. Golan. Performance bounds for orthogonal oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 10(3):571–582, 1981.

[22] X. Han, F. Y. Chin, H.-F. Ting, and G. Zhang. A new upper bound on 2d online bin packing. In *CoRR, abs/0906.0409, 2009*, 2009.

[23] X. Han, K. Iwama, D. Ye, and G. Zhang. Strip packing vs. bin packing. In *AAIM: Proc. 3rd Conference on Algorithmic Aspects in Information and Management*, pages 358–367, 2009.

[24] J. Hurink and J. Paulus. Online scheduling of parallel jobs on two machines is 2-competitive. *Operations Research Letters*, 36(1):51–56, 2008.

[25] J. L. Hurink and J. J. Paulus. Improved online algorithms for parallel job scheduling and strip packing. *Theoretical Computer Science*, doi:10.1016/j.tcs.2009.05.033, 2009.

[26] K. Jansen, L. Prädel, and U. M. Schwarz. Two for one: Tight approximation of 2d bin packing. In *WADS: Proc. 20th Workshop on Algorithms and Data Structures*, pages 399–410, 2009.

[27] K. Jansen and R. Solis-Oba. Rectangle packing with one-dimensional resource augmentation. *Discrete Optimization*, 6(3):310–323, 2010.

[28] K. Jansen and R. Thöle. Approximation algorithms for scheduling parallel jobs: Breaking the approximation ratio of 2. In *ICALP: Proc. 35rd International Colloquium on Automata, Languages and Programming*, pages 234–245, 2008.

[29] K. Jansen and R. van Stee. On strip packing with rotations. In *STOC: Proc. 37th ACM Symposium on Theory of Computing*, pages 755–761, 2005.

[30] K. Jansen and G. Zhang. Maximizing the total profit of rectangles packed into a rectangle. *Algorithmica*, 47(3):323–342, 2007.

[31] B. Johannes. Scheduling parallel jobs to minimize the makespan. *Journal of Scheduling*, 9(5):433–452, 2006.

[32] D. S. Johnson. *Near Optimal Bin Packing Algorithms*. PhD thesis, Massachusetts Institute of Technology, Department of Mathematics, 1973.

[33] N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin packing problem. In *FOCS: Proc. 23rd IEEE Symposium on Foundations of Computer Science*, pages 312–320, 1982.

[34] C. Kenyon and E. Rémila. A near optimal solution to a two-dimensional cutting stock problem. *Mathematics of Operations Research*, 25(4):645–656, 2000.

[35] W. Kern and J. J. Paulus. A tight analysis of Brown-Baker-Katseff sequences for online strip packing. manuscript submitted, 2010.

[36] J. Y.-T. Leung, T. W. Tam, C. S. Wong, G. H. Young, and F. Y. Chin. Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3):271–275, 1990.

[37] K. Li and K. H. Cheng. Static job scheduling in partitionable mesh connected systems. *Journal of Parallel and Distributed Computing*, 10(2):152–159, 1990.

[38] A. Meir and L. Moser. On packing of squares and cubes. *Journal of Combinatorial Theory*, 5(2):126–134, 1968.

[39] I. Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *ESA: Proc. 2nd European Symposium on Algorithms*, pages 290–299, 1994.

[40] S. S. Seiden. On the online bin packing problem. In *ICALP: Proc. 28th International Colloquium on Automata, Languages and Programming*, pages 237–248, 2001.

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

[42] D. D. Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Information Processing Letters*, 10(1):37–40, 1980.

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

[44] R. van Stee. An approximation algorithm for square packing. *Operations Research Letters*, 32(6):535–539, 2004.

[45] A. van Vliet. An improved lower bound for on-line bin packing algorithms. *Information Processing Letters*, 43(5):277–284, 1992.

[46] D. Ye, X. Han, and G. Zhang. A note on online strip packing. *Journal of Combinatorial Optimization*, 17(4):417–423, 2009.

[47] G. Zhang. A 3-approximation algorithm for two-dimensional bin packing. *Operations Research Letters*, 33(2):121–126, 2005.

EIDESSTATTLICHE VERSICHERUNG

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, Juli 2010

Rolf Harren