INSTITUT FÜR INFORMATIK

Reverse-Fit: Ein approximativer Algorithmus für das Strip-Packing-Problem

Lars Prädel

Bericht Nr. 0806
Dezember 2008

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

KIEL

Institut für Informatik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Reverse-Fit: Ein approximativer Algorithmus für das Strip-Packing-Problem

Lars Prädel

Bericht Nr. 0806
Dezember 2008

e-mail: lap@informatik.uni-kiel.de

Dieser Bericht enthält die Studienarbeit des Verfassers

Reverse-Fit: Ein approximativer Algorithmus für das Strip-Packing-Problem

Lars Prädel
Theorie der Parallelität
Institut für Informatik
Christian-Albrechts-Universität zu Kiel

Zusammenfassung

In dieser Arbeit wird ein approximativer levelorientierter Algorithmus namens Reverse-Fit für das Strip-Packing-Problem vorgestellt und analysiert. Dieser Algorithmus basiert auf einer Arbeit von Ingo Schiermeyer. Bei diesem Problem handelt es sich darum eine Menge von Rechtecken in einen Streifen zu packen, so dass eine minimale Packungshöhe entsteht. Hierbei sind die Rechtecke achsenparallel angeordnet und dürfen nicht rotiert werden. Sei OPT die Höhe einer optimalen Packung für eine Instanz L und sei $\mathrm{RF}(L)$ die Höhe der Packung die Reverse-Fit erzeugt. Das Ergebnis der hier vorgestellten Analyse ist $\mathrm{RF}(L) \leq 2 \cdot OPT$.

1 Einleitung

In dieser Arbeit wird ein approximativer Algorithmus für zweidimensionales Strip-Packing namens Reverse-Fit nach Ingo Schiermeyer behandelt [4].
Bei dem zweidimensionalen Strip-Packing-Problem wird eine Instanz, bestehend aus einer Menge von Rechtecken, einem Zielbereich zugeordnet. Dieser Zielbereich entspricht einem Streifen, in dem die Rechtecke gepackt werden, so dass die Rechtecke sich nicht überlappen und eine möglichst geringe Packungshöhe entsteht. Hierbei wird angenommen, dass die Rechtecke achsenparallel angeordnet sind und nicht rotiert werden können. Ferner ist die Breite des Streifens auf 1 gesetzt und somit die Breite der Rechtecke auf 1 beschränkt. Dieses Problem ist eine Weiterführung des eindimensionalen Bin-Packing-Problems, wobei die Höhen der Rechtecke alle gleich sind. Das eindimensionale Bin-Packing-Problem gilt als $\mathcal{NP}$-vollständig, somit ist das zweidimensionale Strip-Packing-Problem auch $\mathcal{NP}$-vollständig und es existiert kein polynomieller Algorithmus der dieses Problem für jede Instanz mit einer optimalen Packungshöhe löst, außer $\mathcal{P} = \mathcal{NP}$.
Ein approximativer Algorithmus für ein Problem liefert eine nicht optimale Lösung, sondern kommt nur bis zu einem bestimmten Faktor an die Lösung heran. Dieser Faktor wird als Güte bezeichnet. Der hier betrachtete approximative Algorithmus hat eine Güte von 2, d.h. der Algorithmus packt die Rechtecke höchstens doppelt so hoch wie in einer optimalen Packung.
Für dieses Problem gibt es mehrere approximative Packungsalgorithmen, darunter die levelorientierten Algorithmen Next-Fit und First-Fit [1]. Bei diesen Algorithmen werden

die Rechtecke in sogenannten Level gepackt, d.h. die Rechtecke werden aufgeteilt und die Rechtecke in den Teilmengen werden separat gepackt. Anschliessend werden diese Level nur noch übereinander gestapelt.
Bei dem Algorithmus Next-Fit werden die Rechtecke von links nach rechts gepackt, bis das nächste bezüglich der Breite nicht mehr hineinpasst. Damit ist dieses Level abgeschlossen und das Rechteck wird dann in das nächste Level gepackt. Bei dem Algorithmus First-Fit werden die Rechtecke auch von links nach rechts gepackt, allerdings in das niedrigste Level, in das dieses Rechteck hineinpasst.
Man kann die Rechtecke als Vorverarbeitungsschritt der Höhe nach absteigend sortieren und erhält dann bei Next-Fit eine Güte von $2 \cdot \mathrm{OPT} + h_{\max}$ und für First-Fit eine Güte von $1{,}7 \cdot \mathrm{OPT} + h_{\max}$, wobei OPT die Höhe einer optimalen Packung ist und $h_{\max}$ die Höhe des höchsten Rechtecks in der Instanz.
Neben diesen eher einfachen Algorithmen existiert noch ein Algorithmus von Sleator, der eine Güte von $2 \cdot \mathrm{OPT} + \frac{1}{2}h_{\max}$ hat, ein APTAS von Jansen und Solis-Oba, und ein AFPTAS von Kenyon und Rémila [5],[2],[3]. Bei dem letzteren Algorithmus kommt man beliebig dicht bis auf einen additiven Term an eine optimale Lösung heran, d.h. dieser Algorithmus hat eine Güte von $(1+\epsilon) \cdot \mathrm{OPT} + O(\frac{1}{\epsilon^2})$ für ein $\epsilon > 0$. Allerdings wird die Laufzeit schlechter, je besser die Güte wird, nämlich $O(n \log n + \log^3 n\epsilon^{-6} \log^3 \epsilon^{-1})$. Also je geringer der Wert von ϵ ist, desto besser ist die Güte, aber desto schlechter ist die Laufzeit. Bei dem Algorithmus von Jansen und Solis-Oba kommt man auch beliebig dicht an eine optimale Lösung heran mit additiver Konstante 1, allerdings ist dieses Approximationsschema nicht vollständig, also die Laufzeit ist nicht polynomiell in $\frac{1}{\epsilon}$. Reverse-Fit hat eine Laufzeit von $O(nlog(n))$, die durch einen Sortierschritt hervorgerufen wird und ist damit für schnellere Berechnungen dem APTAS und dem AFPTAS vorzuziehen.
Die Ausarbeitung gliedert sich wie folgt:
Im Abschnitt 2 werden einige Begriffe erläutert und Abkürzungen eingeführt, die in den folgenden Kapiteln verwendet werden. Abschnitt 3 enthält die Beschreibung des Algorithmus Reverse-Fit und eine Laufzeitanalyse. Die Güte von Reverse-Fit wird in Abschnitt 4 bestimmt. Dabei wird gezeigt, dass der Streifen zur Hälfte gefüllt ist und somit Reverse-Fit eine Güte von 2 hat. Hierfür wird für jedes Level gezeigt, dass es eine Fläche gibt, die die Hälfte dieses Levels ausfüllt. Dabei werden die Flächen für die jeweiligen Level in Unterabschnitten definiert und analysiert. In Abschnitt 5 wird eine Instanz angegeben, für die Reverse-Fit beliebig dicht an diese Güte herankommt und somit gezeigt, dass für diesen Algorithmus keine bessere Analyse existiert. Es folgt daraufhin eine kurze Diskussion in Abschnitt 6.

2 Grundlegendes

Es werden im Folgenden einige grundlegende Begriffe definiert, die für die Beschreibung und Analyse des Algorithmus hilfreich sind. Zunächst wird definiert was eine Instanz für das Problem Strip-Packing ist, daraufhin wird definiert wie ein Rechteck in den Streifen gepackt wird.
In dem Algorithmus Reverse-Fit wird levelweise gepackt, also bietet es sich an die Rechtecke in einem Level fest zu bezeichnen. In der Analyse werden Teilflächen dieser Level verwendet, die aufgrund der festen Bezeichnungen der Rechtecke einfach definiert werden können und an denen genau die Positionen dieser Flächen erkennbar ist.
Sei L im Folgenden eine Instanz für Reverse-Fit, d.h. L besteht aus einer Liste von Rechtecken $r_1, \ldots, r_n$.

Jedes Rechteck r_i hat eine Höhe $h(r_i)$ und eine Breite $b(r_i)$. Die Fläche des Rechtecks ist definiert als $f(r_i) := h(r_i) \cdot b(r_i)$,
Eine Packung P ordnet jedem Rechteck r_i ein Tupel $P(r_i) := (x_i,x'_i,y_i,y'_i)$ zu, wobei die untere linke Ecke des Rechtecks auf die Koordinate (x_i,y_i) gepackt wurde. Es gilt $x'_i := x_i + b(r_i)$ und $y'_i := y_i + h(r_i)$. Diese Zuordnung sei als gepacktes Rechteck bezeichnet.
Der Algorithmus Reverse-Fit packt die Rechtecke levelweise. Sei $N+1$ die Anzahl der Level die Reverse-Fit zu einer gegebenen Instanz erzeugt. Die Level seien mit $L_0,\ldots,L_N$ bezeichnet.
O.B.d.A seien die Rechtecke in dem Level L_i als $r_{l_{i-1}+1},\ldots,r_{l_i}$ für jedes $i \in \{1,\ldots,N\}$ und für L_0 als $r_1,\ldots,r_{l_0}$ bezeichnet, weiter sei R_i die Menge der gepackten Rechtecke in dem Level L_i mit

$$R_i := \{P\left(r_{l_{i-1}+1}\right),\ldots,P(r_{l_i})\}$$

für $i \in \{1,\ldots,N\}$ und für $R_0 := \{P(r_1),\ldots,P(r_{l_0})\}$.
In dem Beweis zur Güte werden häufig Teilflächen betrachtet, also eine Fläche die mit Rechtecken geschnitten wird. Deswegen bieten sich die folgenden Definitionen an:
Der Schnitt einer Fläche $F := (x_F,x'_F,y_F,y'_F)$ mit einem Rechteck r_i wird mit

$$F \cap P(r_i) := (\max(x_F,x_i), \min(x'_F,x'_i), \max(y_F,y_i), \min(y'_F,y'_i))$$

wenn $x_i \leq x_F < x'_i$ oder $x_F \leq x_i < x'_F$ und $y_i \leq y_F < y'_i$ oder $y_F \leq y_i < y'_F$ gilt und $\emptyset$ sonst bezeichnet.
Der Schnitt einer Fläche F mit einer Menge von Rechtecken wird Komponentenweise ausgeführt.
Also bezeichne

$$\begin{aligned} R_i(x,x',y,y') &:= R_i \cap (x,x',y,y') \\ &= \{P\left(r_{l_{i-1}+1}\right),\ldots,P(r_{l_i})\} \cap (x,x',y,y') \\ &= \{P\left(r_{l_{i-1}+1}\right) \cap (x,x',y,y'),\ldots,P(r_{l_i}) \cap (x,x',y,y')\} \end{aligned}$$

Der Flächeninhalt einer Menge A von Rechtecken sei mit $f(A)$ bezeichnet und ist die Summe der Flächeninhalte der einzelnen Komponenten.

3 Der Algorithmus

Der Algorithmus Reverse-Fit ist im wesentlichen ein modifizierter Next-Fit Algorithmus, bei dem die Level nicht nur an der Wand des Streifens beginnen können, sondern auch neben einem Rechteck aus dem ersten Level. Die Besonderheit des Algorithmus liegt darin, dass das zweite Level, das sogenannte rückwärtsgerichtete Level, von rechts nach links gepackt wird und dann abgesenkt wird, bis Rechtecke aus dem ersten Level berührt werden. Die genaue Beschreibung des Algorithmus ist:

RF(L):

1. Stapel die Rechtecke, die eine Breite $> \frac{1}{2}$ haben vom Boden aus. Die Menge dieser Rechtecke wird als nulltes Level bezeichnet und die Summe der Höhen dieser Rechtecke wird mit H_0 bezeichnet.

2. Ordne die restlichen Rechtecke absteigend nach ihrer Höhe.

3. Packe nun diese Rechtecke von links nach rechts auf Höhe H_0 nebeneinander, bis das nächste Rechteck nicht mehr hinein passt. Dies bildet das erste Level.

4. Bei dem zweiten Level werden die Rechtecke von rechts nach links gepackt, so dass die Oberkanten der Rechtecke auf Höhe $H_0 + 2 \cdot h_{\max}$ sind, wobei $h_{\max}$ die Höhe des höchsten Rechtecks ist.
   Überprüfe dabei zunächst für jedes Rechteck, ob es noch in das erste Level hinein passt; dann überprüfe, ob die Summe der Breiten der Rechtecke in dem zweiten Level $> \frac{1}{2}$ ist. Wenn nicht, dann packe das Rechteck neben der rechten Wand oder falls schon Rechtecke in das zweite Level gepackt wurden links neben das letzte Rechteck aus dem zweiten Level.

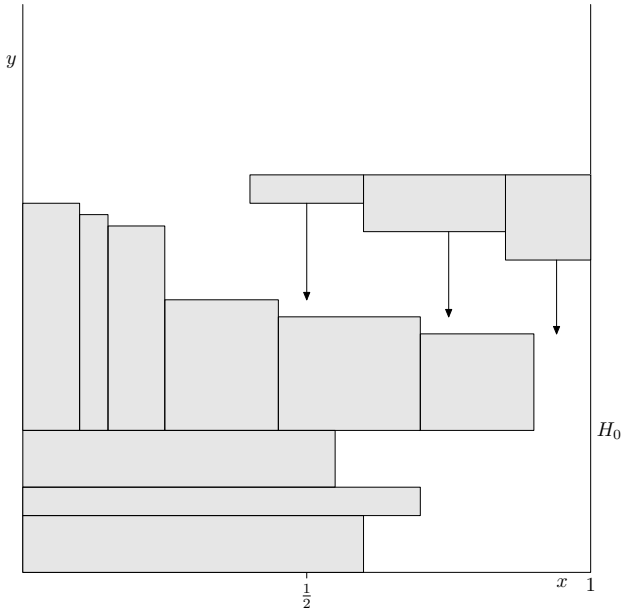

Abbildung 1: Absenken

5. Senke nun das zweite Level ab, bis sich mindestens ein Rechteck aus dem zweiten Level mit einem Rechteck aus dem ersten Level berühren, also die Höhe der Unterkante eines Rechtecks aus dem zweiten Level gleich der Höhe der Oberkante eines Rechtecks aus dem ersten Level ist (siehe Abbildungen 1,2). Das Paar von berührenden Rechtecken, das am weitesten rechts liegt wird mit r_p und r_q bezeichnet, wobei r_p im ersten Level und r_q im zweitem Level ist. Setze zusätzlich $r_k := r_q$. Die Höhe der Oberkante der Rechtecke aus dem zweiten Level wird mit H_1 bezeichnet.

6. Berechne $m_1 := \max(x_p, x_q)$ und $m_2 := \min\left(x'_p, x'_q\right)$

7. Wenn $m_2 \geq \frac{1}{2}$, dann gehe zum nächsten Punkt, ansonsten ist r_q das letzte Rechteck aus dem zweiten Level (siehe Abbildung 3). Senke nun die Rechtecke aus dem zweiten Level bis auf das letzte weiter ab, bis sich wieder Rechtecke aus dem ersten Level mit Rechtecken aus dem zweiten Level berühren (siehe Abbildung 4). Bestimme wieder das am weitesten rechts gelegene, sich berührende Paar und bezeichne diese mit r_p und r_q und berechne m_1 und m_2 wie oben. Ferner bezeichne die Höhe der Oberkante

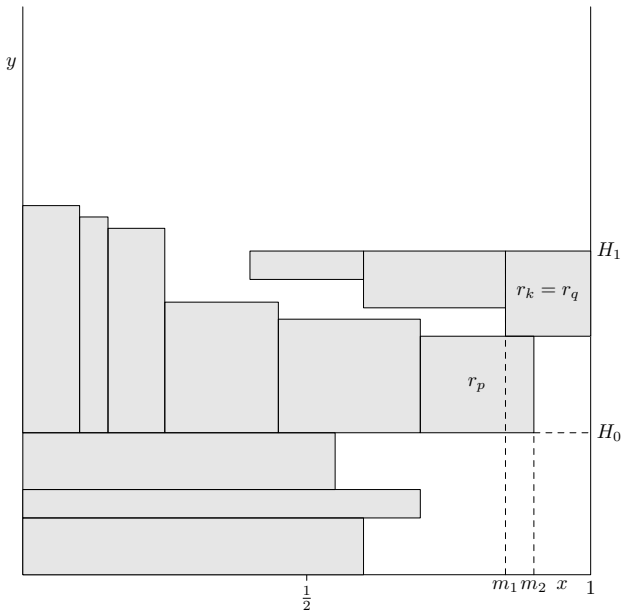

Abbildung 2: Berühren von r_p und r_q

des letzten Rechtecks r_k mit H_2. Der Wert H_1 wird auf die Höhe der Oberkanten der restlichen Rechtecke gesetzt.

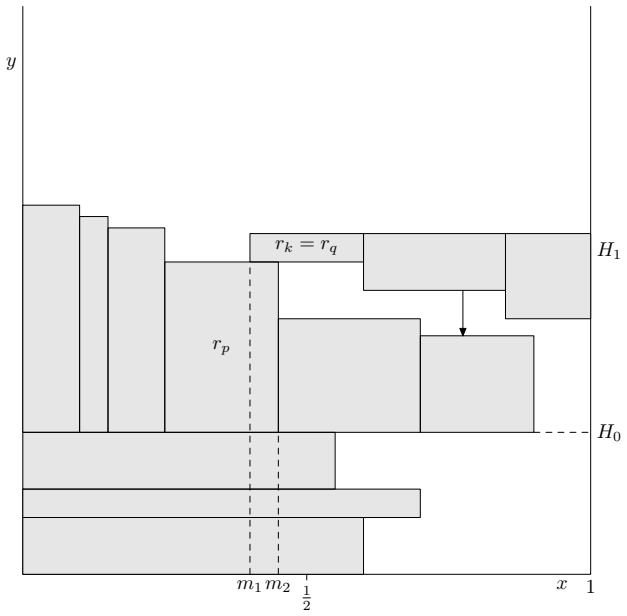

Abbildung 3: $m_2 < \frac{1}{2}$;weiter Absenken

Wenn $H_2 - H_1 \geq h(r_k)$ (siehe Abbildung 5), dann beginnt das dritte Level auf dem vorletztem Rechteck im zweiten Level und das Rechteck r_k wird in das dritte Level gepackt (siehe Abbildung 6), indem es auf Höhe H_1 soweit nach links geschoben wird, bis die linke Kante entweder die linke Wand oder ein Rechteck aus dem ersten Level berührt (siehe Abbildung 7). In diesem Fall wird wieder $H_2 := H_1$ und $r_k := r_q$ gesetzt.

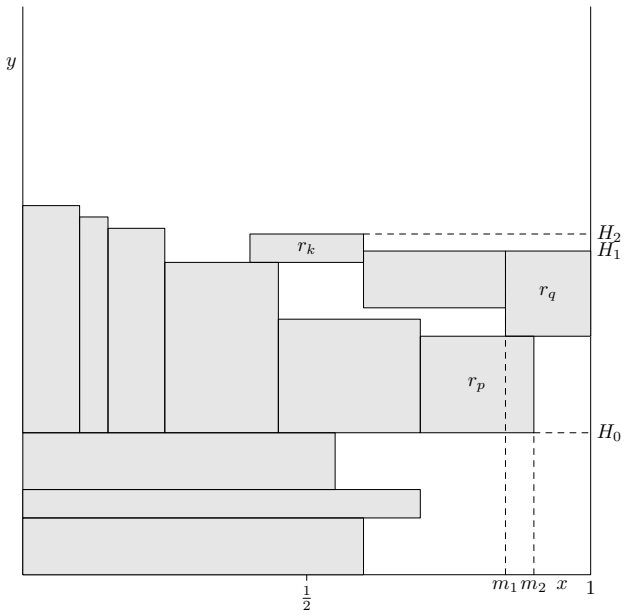

Abbildung 4: r_p,r_q berühren sich

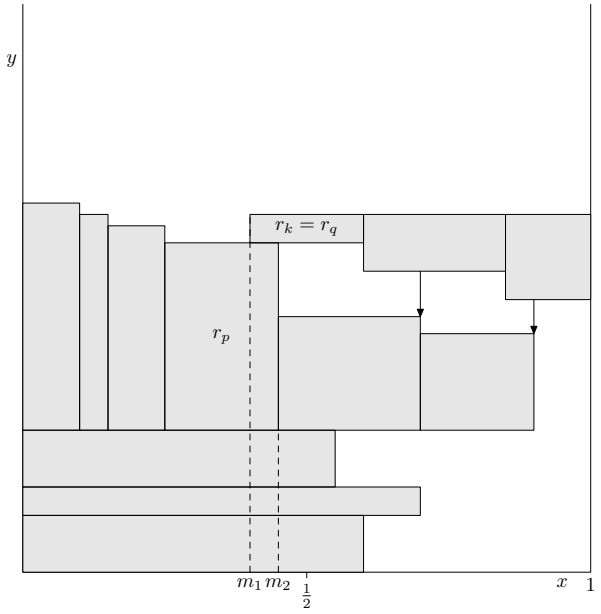

Abbildung 5: $m_2 < \frac{1}{2}$; weiter absenken

8. Berechne $v_p := \max\left(\frac{1}{2}, m_1\right)$

9. Die restlichen Rechtecke werden mit Next-Fit gepackt. Das dritte Level beginnt auf der Höhe H_2, ansonsten wird die Grundlinie des nächsten Levels durch die Oberkante des ersten Rechtecks aus dem vorherigen Level definiert. Die Level müssen nicht unbedingt an der linken Wand anfangen, sondern können auch von der rechten Kante eines Rechtecks aus dem ersten Level beginnen.

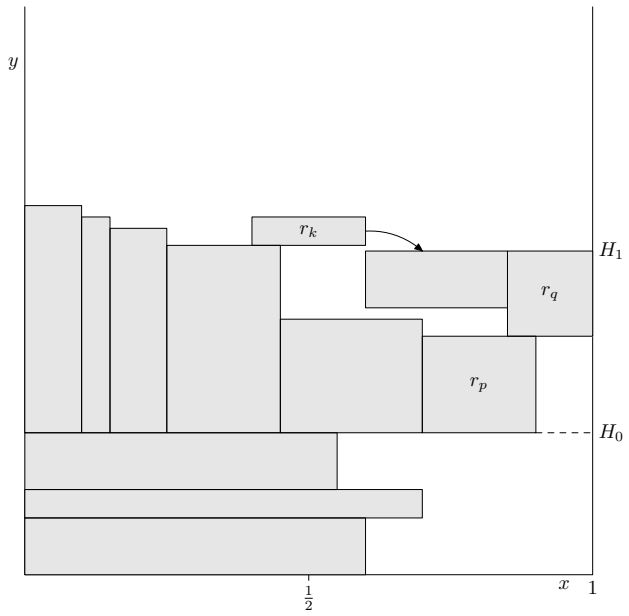

Abbildung 6: $H_2 - H_1 \geq h(r_k)$

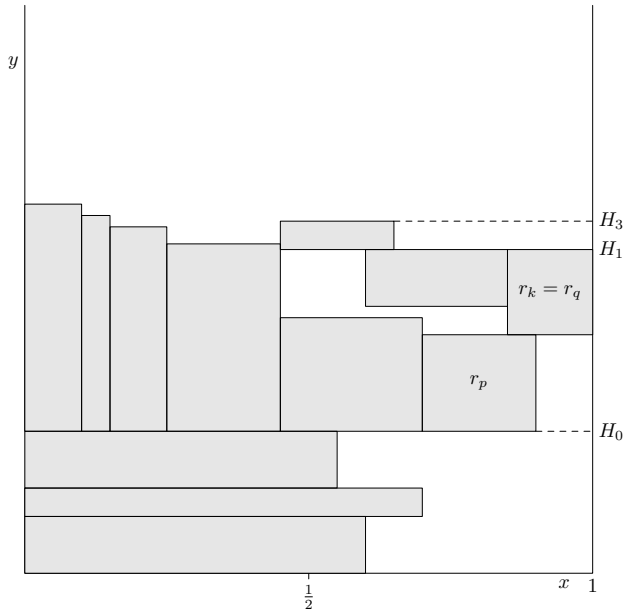

Abbildung 7: Verschieben von r_k

Laufzeit Das Aussortieren der breiten Rechtecke kann in $O(n)$ ausgeführt werden, der Sortierschritt in $O(n \log n)$. Die weiteren Schritte, also zunächst die ersten beiden Level zu packen, sowie die weiteren Level zu packen, liegt in $O(n)$. Das Absenken kann so implementiert werden, dass die Rechtecke der ersten beiden Level in eine Liste gespeichert werden, ferner wird der Index von dem letzten Rechteck des ersten Levels und des ersten Rechtecks aus dem zweiten Level abgespeichert. Von diesen Indizes ausgehend werden die Höhen dieser Rechtecke addiert und abgespeichert. Wenn die linke Kante des Rechtecks im ersten Level weiter links liegt wird der Index für das zweite Level inkrementiert, ansonsten wird der Index aus dem ersten Level dekrementiert. Daraufhin werden wieder die Höhen dieser Rechtecke addiert, wenn die Summe grösser ist als der zuvor gespeicherte Wert wird

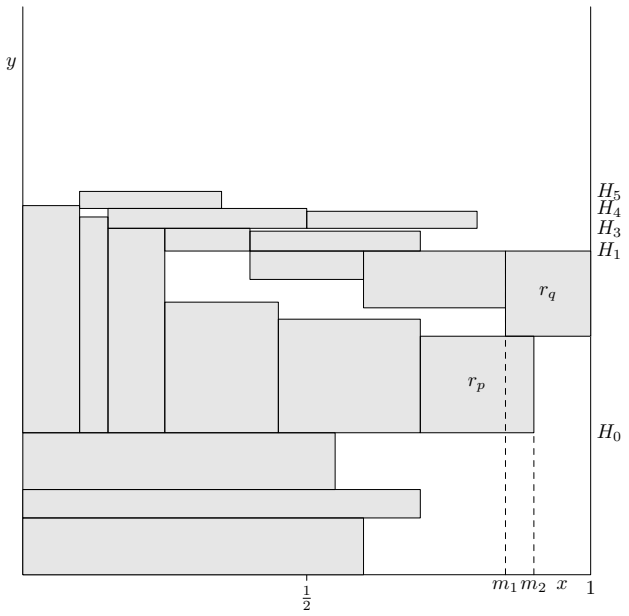

Abbildung 8: Alle Level

dieser überschrieben. Dies wird solange iteriert, bis keine Rechtecke mehr übereinander liegen. Die gespeicherte Summe der Höhen ist dann die Höhe der Oberkante der Rechtecke des zweiten Levels und die dazugehörigen Rechtecke sind r_p und r_q. Es entstehen dadurch höchstens n Vergleiche, also liegt auch das Absenken in $O(n)$. Die Gesamtlaufzeit liegt somit in $O(n \log n)$.

4 Die Analyse

Im Folgenden wird die Güte des Algorithmus analysiert. Hierbei wird gezeigt, dass mindestens die Hälfte des Streifens gefüllt ist. Dazu wird für jedes Level eine Fläche definiert, die für die Höhe dieses Levels die Hälfte des Streifens ausfüllt.

Satz 1. *Sei $RF(L)$ die Höhe der Packung von Reverse-Fit ausgeführt mit einer Instanz L und OPT die Höhe einer optimalen Packung von L. Dann gilt*

$$RF(L) \leq 2 \cdot OPT$$

Zunächst die trivialen Fälle: Seien nur Rechtecke in das nullte und erste Level gepackt worden, dann ist die Höhe von $\mathrm{RF}(L) = H_0 + h_{\max} \leq \mathrm{OPT} + \mathrm{OPT}$. Seien also $N \geq 2$ Level gepackt worden. Ziel ist es für jedes Level $L_0, \ldots, L_N$ mit Höhen $h_0, \ldots, h_N$ disjunkte Flächen $A_0, \ldots, A_N$ zu finden in denen Rechtecke gepackt wurden und für die gilt $f(A_i) \geq \frac{1}{2} h_i$, für $i \in \{0, \ldots, N\}$.
Daraus folgt, da die Breite des Streifens 1 ist, dass

$$\mathrm{OPT} \geq \sum_{i=1}^{n} f(r_i) \geq \sum_{j=0}^{N} f(A_j) \geq \frac{1}{2} \sum_{j=0}^{N} h_j = \frac{1}{2} \mathrm{RF}(L).$$

Definiere die Flächen wie folgt (siehe Abbildung 9):

$$
\begin{aligned}
A_0 :=& R_0 \\
A_1 :=& R_1\left(0,v_p,H_0,H_0+h\left(r_p\right)\right) \cup R_2\left(v_p,1,H_0+h\left(r_p\right),H_1\right) \\
A_2 :=& R_1\left(0,1,H_0+h(r_p),H_2-h\left(r_k\right)\right) \cup R_2\left(x_k,\frac{1}{2},H_1,H_2\right) \\
A_3 :=& R_1\left(v_p,1,H_0,H_0+h\left(r_p\right)\right) \cup R_1\left(0,1,H_2-h\left(r_k\right),H_2\right) \\
& \cup R_2\left(\frac{1}{2},v_p,H_0+h\left(r_p\right),H_2\right) \cup R_3\left(0,\frac{1}{2},H_2,H_3\right) \\
A_i :=& R_1\left(0,1,H_{i-2},H_{i-1}\right) \cup R_{i-1}\left(\frac{1}{2},1,H_{i-2},H_{i-1}\right) \cup R_i\left(0,\frac{1}{2},H_{i-1},H_i\right) \\
& \text{für } i \in \{4,\ldots,N\}
\end{aligned}
$$

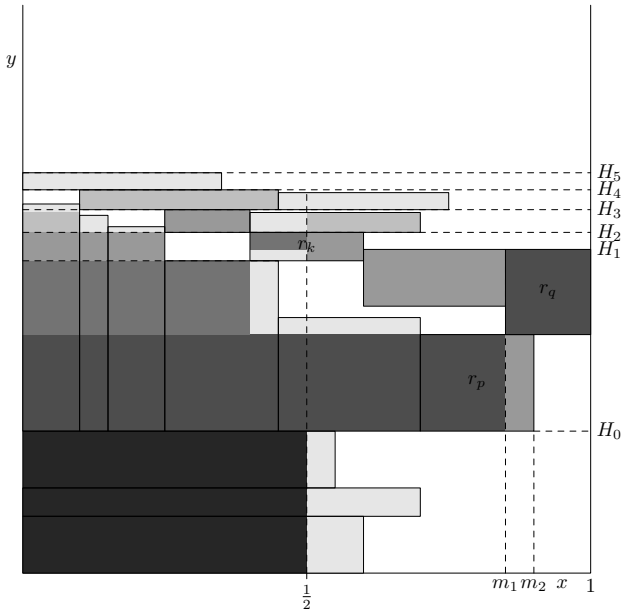

Abbildung 9: Die Flächen im Überblick

Dabei muss beachtet werden, dass auch $A_2 = \emptyset$ auftreten kann und zwar genau dann, wenn $H_2 = H_1$ ist. In diesem Fall ist $r_k = r_q$, also $H_2 - h\left(r_k\right) = H_1 - h\left(r_q\right) = H_0 + h\left(r_p\right)$.

4.1 Die Fläche A_0

Für die Fläche A_0 werden die Rechtecke aus dem nullten Level verwendet. In diesem Level sind alle Rechtecke die breiter sind als $\frac{1}{2}$. Also ist der Streifen bis zur Höhe H_0 bis zu Hälfte gefüllt(siehe Abbildung 10):

$$
f\left(A_0\right) = f\left(R_0\right) = \sum_{i=1}^{l_0} f\left(r_i\right) = \sum_{i=1}^{l_0} b\left(r_i\right) h\left(r_i\right) \geq \sum_{i=1}^{l_0} \frac{1}{2} h\left(r_i\right) = \frac{1}{2} \sum_{i=0}^{l_0} h\left(r_i\right) = \frac{1}{2} h_0.
$$

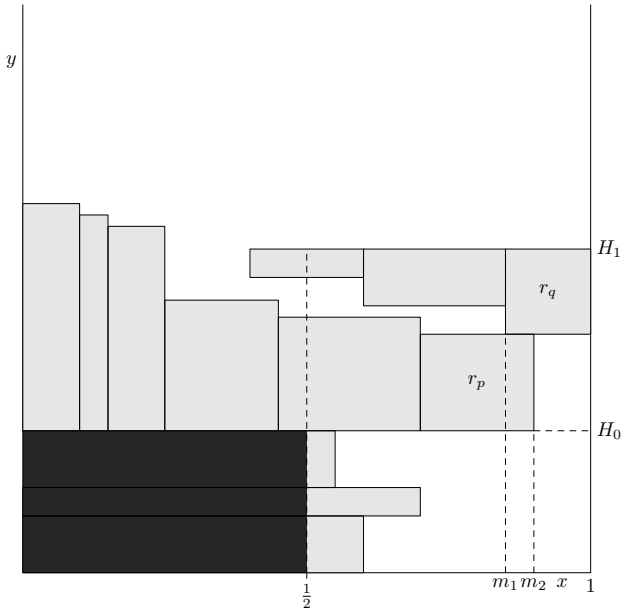

Abbildung 10: A_0

4.2 Die Fläche A_1

Für die Fläche A_1, welche die Höhe $h_1 = H_1 - H_0$ abdecken soll, wird eine Teilfläche der Rechtecke von dem ersten Level mit einer Breite $v_p = \max(\frac{1}{2}, m_1)$ und einer Höhe von $h(r_p)$ verwendet und von dem zweiten Level von v_p bis zur Wand des Streifens in einer Höhe von $h(r_q)$. Diese Fläche hat eine Gesamtbreite von 1 und bis zur Breite $\frac{1}{2}$ eine Höhe von $h(r_p)$ und ab $\frac{1}{2}$ eine Höhe von mindestens $h(r_q)$ (siehe Abbildung 11). Da $v_p \geq \frac{1}{2}$ und

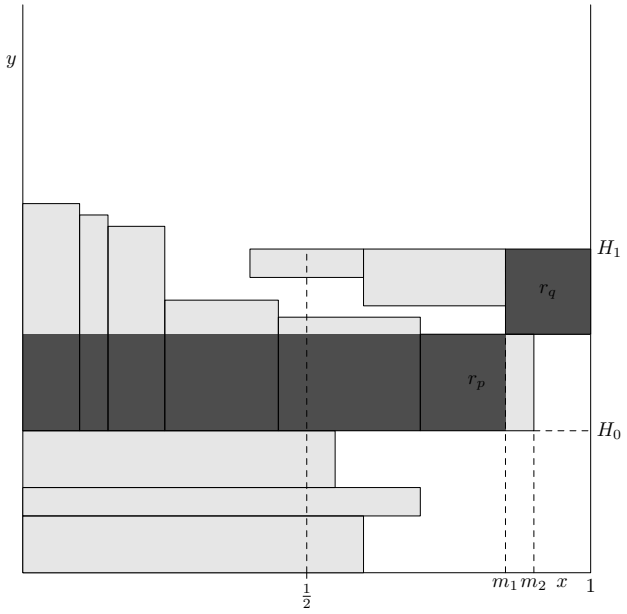

Abbildung 11: A_1

$h(r_p) \geq h(r_q)$ gilt:

$$\begin{aligned}
f(A_1) =& f(R_1(0,v_p,H_0,H_0+h(r_p))) + f(R_2(v_p,1,H_0+h(r_p),H_1)) \\
=& (v_p-0)(H_0+h(r_p)-H_0) + (1-v_p)(H_1-(H_0+h(r_p))) \\
=& v_p \cdot h(r_p) + (1-v_p)h(r_q) \\
=& \frac{1}{2}h(r_p) + \left(v_p - \frac{1}{2}\right)h(r_p) + (1-v_p)h(r_q) \\
\geq& \frac{1}{2}h(r_p) + \left(v_p - \frac{1}{2}\right)h(r_q) + (1-v_p)h(r_q) \\
=& \frac{1}{2}h(r_p) + \left(v_p - \frac{1}{2} + 1 - v_p\right)h(r_q) \\
=& \frac{1}{2}h(r_p) + \frac{1}{2}h(r_q) \\
=& \frac{1}{2}(h(r_p) + h(r_p)) \\
=& \frac{1}{2}h_1.
\end{aligned}$$

4.3 Die Fläche A_2

Wenn $A_2 = \emptyset$ gilt ist nichts zu zeigen, da dann auch $H_1 = H_2$ ist und somit $h_2 = H_2 - H_1 = 0$ ist. Sei also $A_2 \neq \emptyset$ und damit $H_1 \neq H_2$, also wurde ein zweites Mal abgesenkt. Diese Distanz ist nicht so hoch wie das letzte Rechteck in dem zweiten Level. Also das letzte und das vorletzte Rechteck aus dem zweiten Level berühren sich nach dem weiteren Absenken noch und somit wird das letzte Rechteck aus diesem Level nicht in das dritte Level verschoben (siehe Abbildung 12). Es gilt:

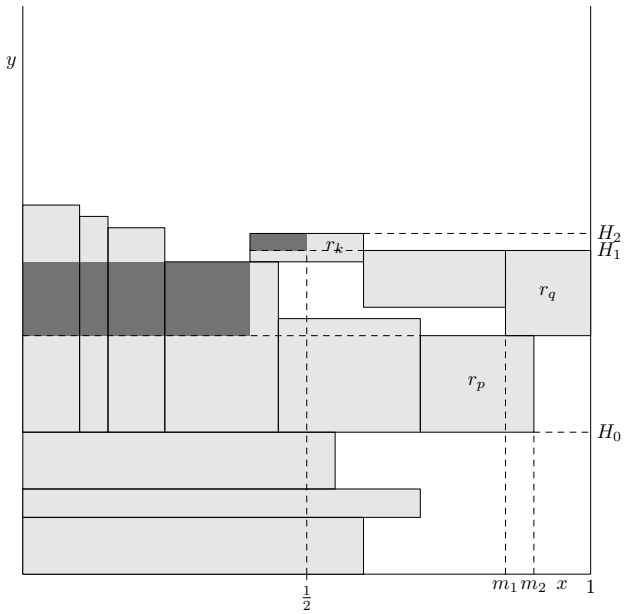

Abbildung 12: A_2

$$
\begin{aligned}
f(A_2) =& f(R_1(0{,}1{,}H_0 + h(r_p){,}H_2 - h(r_k))) + f\left(R_2\left(x_k{,}\frac{1}{2}{,}H_1{,}H_2\right)\right) \\
\geq& (x_k - 0)(H_2 - h(r_k) - (H_0 + h(r_p))) + \left(\frac{1}{2} - x_k\right)(H_2 - H_1) \\
=& x_k(H_2 - h(r_k) - (H_0 + h(r_p))) + \left(\frac{1}{2} - x_k\right)(H_2 - H_1) \\
\geq& x_k(H_2 - h(r_q) - (H_0 + h(r_p))) + \left(\frac{1}{2} - x_k\right)(H_2 - H_1) \\
=& x_k(H_2 - (H_0 + h(r_p) + h(r_q))) + \left(\frac{1}{2} - x_k\right)(H_2 - H_1) \\
=& x_k(H_2 - H_1) + \left(\frac{1}{2} - x_k\right)(H_2 - H_1) \\
=& \frac{1}{2}h_2.
\end{aligned}
$$

4.4 Die Fläche A_3

Die Fläche A_3 soll nun die Höhe des dritten Levels abdecken und besteht aus folgenden Teilflächen:

$$
\begin{aligned}
f(A_3) =& f(R_1(v_p{,}1{,}H_0{,}H_0 + h(r_p))) \\
&+ f(R_1(0{,}1{,}H_2 - h(r_k){,}H_2)) \\
&+ f\left(R_2\left(\frac{1}{2}{,}v_p{,}H_0 + h(r_p){,}H_2\right)\right) \\
&+ f\left(R_3\left(0{,}\frac{1}{2}{,}H_2{,}H_3\right)\right)
\end{aligned}
$$

Hierbei treten zwei Fälle auf:

1. Die rechte Kante des ersten Rechtecks in dem dritten Level ist größer als $\frac{1}{2}$ ($x'_{l_2+1} > \frac{1}{2}$) (siehe Abbildung 13).
2. Die rechte Kante des ersten Rechtecks in dem drittem Level ist kleiner gleich $\frac{1}{2}$ ($x'_{l_2+1} \leq \frac{1}{2}$) (siehe Abbildung 14).

Wenn das letzte Rechteck des zweiten Levels in das dritte Level verschoben wurde, ist man in dem ersten Fall (vgl. Abbildung 14).
Für den ersten Fall benötigt man, um die Höhe des dritten Levels abzudecken, die Flächen

- $R_1(v_p{,}1{,}H_0{,}H_0 + h(r_p))$ und
- $R_3\left(0{,}\frac{1}{2}{,}H_2{,}H_3\right)$.

Für den zweiten Fall benötigt man die Flächen

- $R_1(0{,}1{,}H_2 - h(r_k){,}H_2)$,

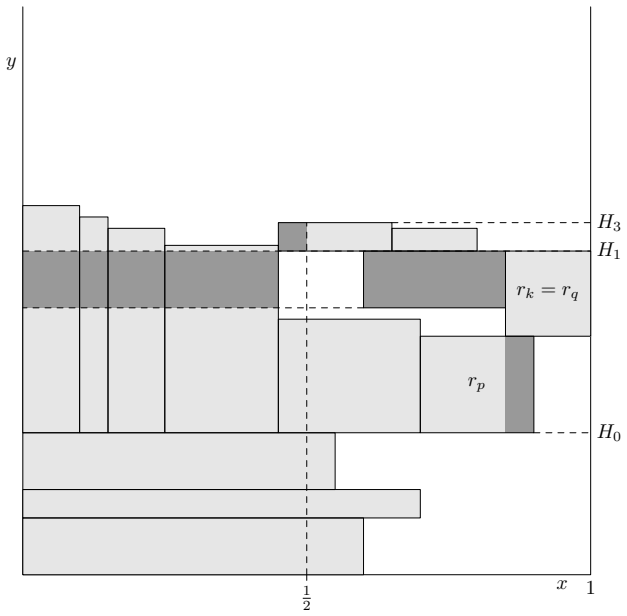

Abbildung 13: A_3, erster Fall

- $R_2\left(\frac{1}{2},v_p,H_0+h\left(r_p\right),H_2\right)$ und
- $R_3\left(0,\frac{1}{2},H_2,H_3\right)$.

Für den ersten Fall gilt dann, dass das erste Rechteck in dem dritten Level ein Rechteck aus dem ersten Level berührt, da die rechte Kante dieses Rechtecks $\frac{1}{2}$ überschreitet, und das Rechteck eine geringere Breite als $\frac{1}{2}$ hat.
Somit gilt:

$$f\left(R_1\left(0,1,H_2-h\left(r_k\right),H_2\right)\right)=x_{l_2+1}\cdot\left(H_2-\left(H_2-h\left(r_k\right)\right)\right)=x_{l_2+1}\cdot h\left(r_k\right)\geq x_{l_2+1}\cdot h(r_{l_2+1}).$$

Es folgt also:

$$\begin{aligned}
f\left(A_3\right)\geq & f\left(R_1\left(0,1,H_2-h\left(r_k\right),H_2\right)\right)+f\left(R_3\left(0,\frac{1}{2},H_2,H_3\right)\right)\\
&\geq x_{l_2+1}\cdot h(r_{l_2+1})+\left(\frac{1}{2}-x_{l_2+1}\right)\cdot h\left(r_{l_2+1}\right)\\
&=\left(x_{l_2+1}+\frac{1}{2}-x_{l_2+1}\right)\cdot h\left(r_{l_2+1}\right)\\
&=\frac{1}{2}\cdot h\left(r_{l_2+1}\right)\\
&=\frac{1}{2}h_3
\end{aligned}$$

Zum zweiten Fall gilt, dass insbesondere nicht der Fall enthalten ist, in dem das letzte Rechteck aus dem zweiten Level in das dritte Level verschoben wurde, da ansonsten die rechte Kante dieses Rechtecks $\frac{1}{2}$ überschritten hätte. Also gilt, dass die Summe der Breiten der Rechtecke aus dem zweiten Level grösser $\frac{1}{2}$ ist. Weiter muss beachtet werden, dass der

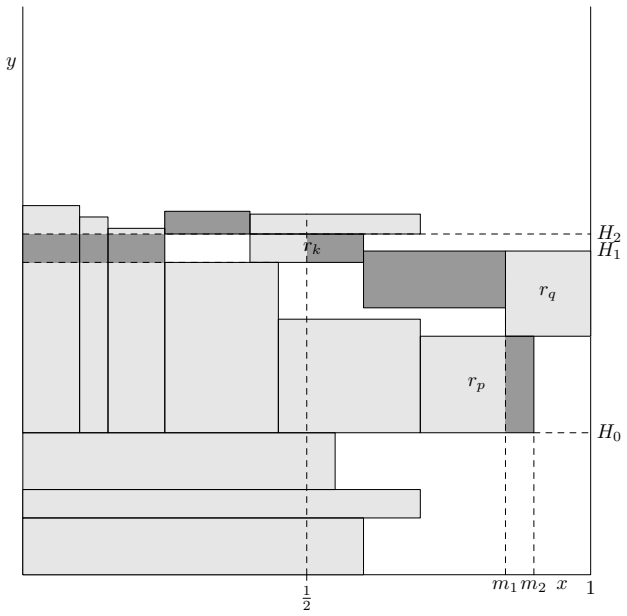

Abbildung 14: A_3, zweiter Fall

Algorithmus überprüft, dass das erste Rechteck im dritten Level nicht mehr in das erste Level hinein passt.
Es gilt also:

$$\begin{aligned} f\left(R_1\left(v_p,1,H_0,H_0+h\left(r_p\right)\right)\right) &\geq \left(x'_{l_1}-v_p\right)\cdot h\left(r_{l_1}\right) \\ &\geq \left(x'_{l_1}-v_p\right)\cdot h\left(r_{l_2+1}\right) \\ &= \left(x'_{l_1}-v_p\right)\cdot h_3, \end{aligned} \tag{1}$$

$$\begin{aligned} f\left(R_2\left(\frac{1}{2},v_p,H_0+h\left(r_p\right),H_2\right)\right) &\geq \left(v_p-\frac{1}{2}\right)\cdot h\left(r_{l_2}\right) \\ &\geq \left(v_p-\frac{1}{2}\right)\cdot h\left(r_{l_2+1}\right) \\ &= \left(v_p-\frac{1}{2}\right)\cdot h_3, \end{aligned} \tag{2}$$

$$\begin{aligned} f\left(R_3\left(0,\frac{1}{2},H_2,H_3\right)\right) &\geq w\left(r_{l_2+1}\right)\cdot h\left(r_{l_2+1}\right) \\ &= w\left(r_{l_2+1}\right)\cdot h_3. \end{aligned} \tag{3}$$

Ausserdem gilt, da das Rechteck r_{l_2+1} nicht mehr in das erste Level hinein passt, dass

$$w\left(r_{l_2+1}\right)+x'\left(l_1\right)>1. \tag{4}$$

Insgesamt folgt:

$$\begin{aligned}
f(A_3) =& f(R_1(v_p,1,H_0,H_0+h(r_p))) \\
&+ f\left(R_2\left(\frac{1}{2},v_p,H_0+h(r_p),H_2\right)\right) + f\left(R_3\left(0,\frac{1}{2},H_2,H_3\right)\right) \\
\overbrace{\geq}^{1,2,3}& \left(x'_{l_1}-v_p\right)\cdot h_3 + \left(v_p-\frac{1}{2}\right)\cdot h_3 + w(r_{l_2+1})\cdot h_3 \\
=& \left(x'_{l_1}-v_p+v_p-\frac{1}{2}+w(r_{l_2+1})\right)\cdot h_3 \\
=& \left(x'_{l_1}+w(r_{l_2+1})-\frac{1}{2}\right)\cdot h_3 \\
\overbrace{\geq}^{4}& \frac{1}{2}\cdot h_3
\end{aligned}$$

4.5 Die Flächen höherer Level

Für die folgenden Level kann eine einheitliche Definition verwendet werden. Dabei wird

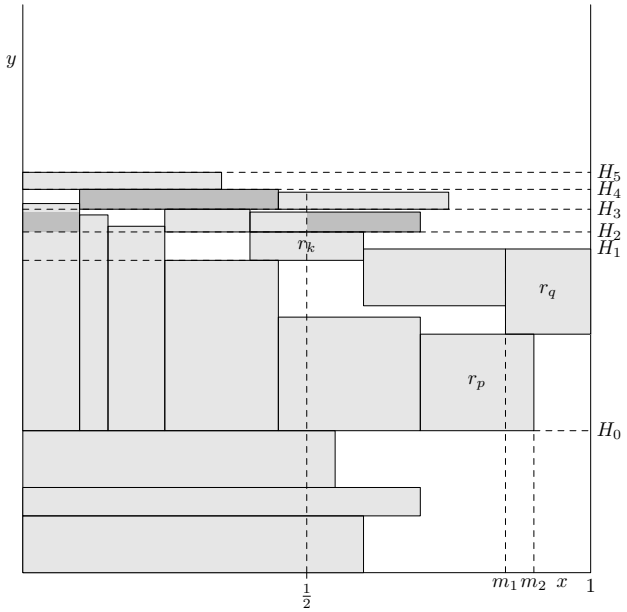

Abbildung 15: Flächen höherer Level

das erste Rechteck aus dem jeweiligen Level bis $\frac{1}{2}$ verwendet und von $\frac{1}{2}$ die Fläche von dem vorherigen Level. Da allerdings die Level auch neben einem Rechteck aus dem ersten Level beginnen können, muss auch von dort die Fläche in der Höhe vom vorherigen Level genommen werden (siehe Abbildung 15).
Es ergibt sich somit:
$f(A_i) = f(R_1(0,1,H_{i-2},H_{i-1})) + f\left(R_{i-1}\left(\frac{1}{2},1,H_{i-2},H_{i-1}\right)\right) + f\left(R_i\left(0,\frac{1}{2},H_{i-1},H_i\right)\right)$
Es gilt:

$$f(R_1(0,1,H_{i-2},H_{i-1})) \geq \left(x_{l_{i-1}+1}-0\right)\cdot h_{i-1} = x_{l_{i-1}+1}\cdot h_{i-1} \geq x_{l_{i-1}+1}\cdot h_i, \qquad (1)$$

$$f\left(R_{i-1}\left(\frac{1}{2},1,H_{i-1},H_i\right)\right) = \left(x'_{l_{i-1}} - \frac{1}{2}\right) \cdot h_{i-1} \geq \left(x'_{l_{i-1}} - \frac{1}{2}\right) \cdot h_i \tag{2}$$

und

$$f\left(R_i\left(0,\frac{1}{2},H_{i-1},H_i\right)\right) \geq \left(\min\left(\frac{1}{2},x'_{l_{i-1}+1}\right) - x_{l_{i-1}+1}\right) \cdot h_i. \tag{3}$$

Da das erste Rechteck aus dem jeweiligen Level nicht mehr in das vorherige passt, gilt

$$x'_{l_{i-1}} + w\left(r_{l_{i-1}+1}\right) > 1. \tag{4}$$

Also:

$$\begin{aligned}
f(A_i) &= f\left(R_1\left(0,1,H_{i-2},H_{i-1}\right)\right) + f\left(R_{i-1}\left(\frac{1}{2},1,H_{i-2},H_{i-1}\right)\right) \\
&\quad + f\left(R_i\left(0,\frac{1}{2},H_{i-1},H_i\right)\right) \\
&\overbrace{\geq}^{1,2,3} x_{l_{i-1}+1} \cdot h_i + \left(x'_{l_{i-1}} - \frac{1}{2}\right) \cdot h_i \\
&\quad + \left(\min\left(\frac{1}{2},x'_{l_{i-1}+1}\right) - x_{l_{i-1}+1}\right) \cdot h_i \\
&= \left(x_{l_{i-1}+1} + x'_{l_{i-1}} - \frac{1}{2} + \min\left(\frac{1}{2},x'_{l_{i-1}+1}\right) - x_{l_{i-1}+1}\right) \cdot h_i \\
&= \min\left(x'_{l_{i-1}} - \frac{1}{2} + \frac{1}{2},x'_{l_{i-1}} - \frac{1}{2} + x'_{l_{i-1}+1}\right) \cdot h_i \\
&\geq \min\left(x'_{l_{i-1}},x'_{l_{i-1}} - \frac{1}{2} + w\left(r_{l_{i-1}+1}\right)\right) \cdot h_i \\
&\overbrace{\geq}^{4} \min\left(x'_{l_{i-1}},1 - \frac{1}{2}\right) \cdot h_i \\
&\geq \frac{1}{2} \cdot h_i.
\end{aligned}$$

Wir haben nun gezeigt, dass für alle Level disjunkte Flächen existieren, die einen Flächeninhalt von $\frac{1}{2}$ mal der Höhe des jeweiligen Levels haben. Somit wurde für Instanzen mit mehr als zwei Leveln gezeigt, dass der Streifen mindestens zur Hälfte gefüllt ist. Dies schließt den Beweis zur Güte.

5 Angabe von Instanzen mit annähernder Zweier-Approximation

Wir können eine Instanz ohne breite Rechtecke angeben, für die Reverse-Fit eine Lösung ausgibt, die beliebig dicht an die Güte von 2 herankommt.

Satz 2. *Es gibt Instanzen L mit $RF(L) \geq \frac{2}{1+\epsilon} \cdot OPT$, für ein $\epsilon > 0$.*

Beweis. Sei $\epsilon > 0$. Setze $\epsilon' \leq \min\left(\frac{1}{4},\frac{\epsilon}{4}\right)$, so dass $\frac{3}{4\epsilon'} - 3$ durch 2 teilbar ist.
Die Instanz L enthält die folgenden Rechtecke:

- Ein Rechteck mit Breite $3\epsilon'$ und Höhe 3,
- drei Rechtecke mit Breite $\frac{1}{2} - \epsilon'$ und Höhe 1,
- $\frac{3}{4\epsilon'} - 3$ mal
  - einem Rechteck mit Breite $\frac{1}{2} - \epsilon'$ und Höhe 1 und
  - drei Rechtecke mit Breite ϵ' und Höhe 1.

Diese Rechtecke haben alle eine Breite kleiner $\frac{1}{2}$ und werden somit nicht mit Reverse-Fit in das nullte Level gepackt. Weiter gilt, dass diese Rechtecke so wie sie in dieser Liste stehen bereits nach der Höhe sortiert sind und können o.B.d.A annehmen, dass der Sortierschritt keine Veränderung an der Reihenfolge der Rechtecke vornimmt.
Zunächst gilt, dass in dem ersten Level die ersten beiden Rechtecke gepackt werden. Die nächsten beiden Rechtecke passen nicht mehr in dieses Level und werden in das zweite Level gepackt. Die Breite in dem zweiten Level ist $1 - 2\epsilon'$, somit berührt das zweite Rechteck das erste Rechteck in dem ersten Level mit Höhe 3. Deswegen wird weiter abgesenkt und das letzte Rechteck wird in das dritte Level verschoben. Die nächsten Level bestehen jeweils aus einem Rechteck mit Breite $\frac{1}{2} - \epsilon'$ und drei Rechtecken mit Breite ϵ' (vgl. Abbildung 16). Die Höhe die Reverse-Fit erzeugt beträgt somit $3 + \frac{3}{4\epsilon'} - 3 = \frac{3}{4\epsilon'}$
Bei einer optimalen Packung werden je zwei Rechtecke der Breite ϵ' aus den höheren Leveln der Reverse-Fit Packung in die ersten drei Level gepackt (vgl. Abbildung 17). In den ersten drei Leveln ist ein freier Platz von $3 \cdot \left(\frac{1}{2} - 2\epsilon'\right) = \frac{3}{2} - 6\epsilon' = 2\epsilon' \cdot \left(\frac{3}{4\epsilon'} - 3\right)$.
Also passen diese Rechtecke in den freien Platz in den ersten drei Leveln. Ab dem vierten Level werden jeweils zwei Rechtecke der Breite $\frac{1}{2} - \epsilon'$ und zwei Rechtecke der Breite ϵ' gepackt. Somit entsteht eine Höhe von $3 + \frac{1}{2} \cdot \left(\frac{3}{4\epsilon'} - 3\right) = \frac{3}{8\epsilon'} - \frac{3}{2} = \frac{3+12\epsilon'}{8\epsilon'}$.
Es folgt

$$\begin{aligned}
\frac{\mathrm{RF}(L)}{\mathrm{OPT}(L)} &= \frac{\frac{3}{4\epsilon'}}{\frac{3+12\epsilon'}{8\epsilon'}} \\
&= \frac{3 \cdot 8\epsilon'}{4\epsilon' \left(3 + 12\epsilon'\right)} \\
&= \frac{6}{3 + 12\epsilon'} \\
&= \frac{2}{1 + 4\epsilon'} \\
&\geq \frac{2}{1 + 4\frac{\epsilon}{4}} \\
&= \frac{2}{1 + \epsilon}
\end{aligned}$$

□

6 Diskussion und Ausblick

Der hier betrachtete Algorithmus Reverse-Fit ist eine Abwandlung des Algorithmus von Ingo Schiermeyer, da hier ein modifizierter Next-Fit-Algorithmus angewendet wird anstelle eine modifizierten First-Fit-Algorithmus. Die Modifikation besteht darin, dass die

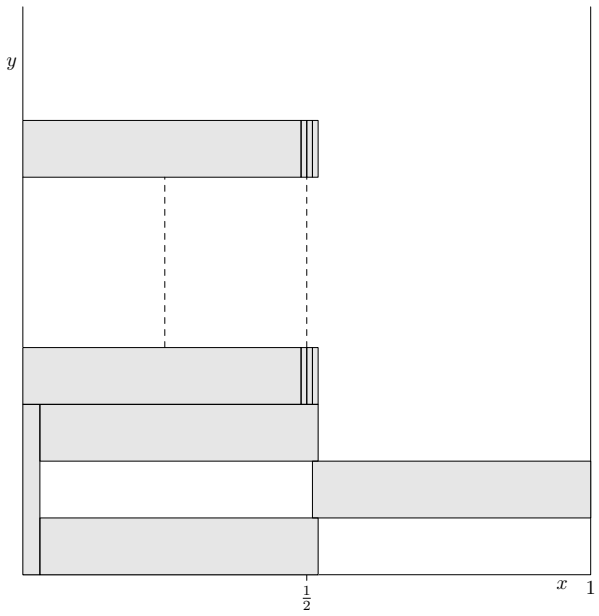

Abbildung 16: Reverse-Fit

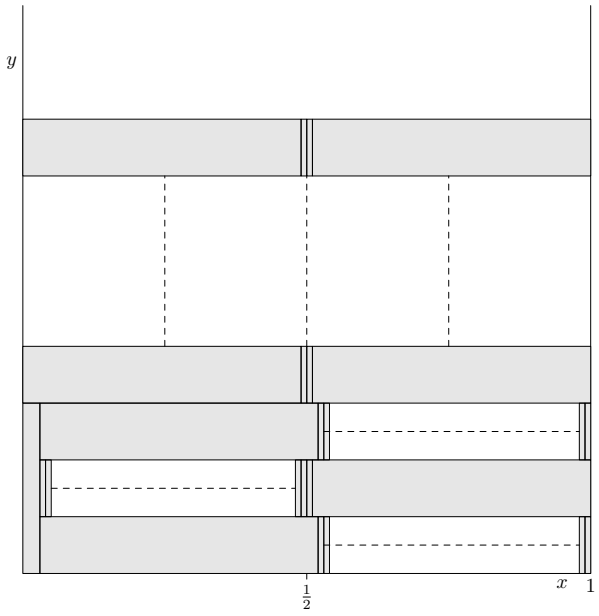

Abbildung 17: optimale Packung

Level nicht an der Wand des Streifens beginnen müssen, sondern auch an der rechten Seite eines Rechtecks aus dem ersten Level beginnen können. Da sich aber der Beweis zur Güte durch diese Abwandlung nicht ändert und eine Erweiterung von dem Beweis der Güte von Next-Fit darstellt, ist diese Abwandlung sinnvoll. Somit kann Reverse-Fit als Erweiterung von Next-Fit angesehen werden, wo sich im Grunde nur das zweite Level, das rückwärtsgerichtete Level, von dem modifizierten Next-Fit Algorithmus unterscheidet.
Durch Verwenden von First-Fit kann ohne weiteres keine Verbesserung der Güte erreicht werden, da schon für das nullte Level keine bessere Güte garantiert wird. Demzufolge müssten die breiten Rechtecke auf die anderen Level aufgeteilt werden, wobei allerdings keine breiten Rechtecke in den ersten beiden Leveln auftreten dürfen und auch das Auf-

teilen in den oberen Leveln stellt Probleme dar. Genauso verhält es sich, wenn mehrere rückwärtsgerichtete Level eingeführt werden. Auch hier stößt man auf Probleme, die durch die breiten Rechtecke hervorgerufen werden. Nur durch eine Restriktion, die keine breiten Rechtecke erlaubt, könnte vielleicht eine bessere Güte analysiert werden.

Literatur

[1] 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 of Computing*, 9(4):808–826, 1980.

[2] K. Jansen and R. Solis-Oba. New approximability results for 2-dimensional packing problems. In *Mathematical Foundations of Computer Science*, volume 4708 of *Lecture Notes in Computer Science*, pages 103–114. Springer, 2007.

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

[4] I. Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *Proceedings of the Second Annual European Symposium on Algorithms*, pages 290–299. Springer-Verlag, 1994.

[5] D. D. K. D. B. Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Information Processing Letters*, 10(37-40), 1980.