



Bin Packing

Prof. Dr. S. Albers
Prof. Dr. Th. Ottmann

1. Problembeschreibung und einfache Beobachtungen
2. Approximative Lösung des Online Bin Packing Problems
3. Approximative Lösung des Offline Bin Packing Problems

Problembeschreibung

Gegeben:

n Objekte der Größen

$$s_1, \dots, s_n$$

mit $0 < s_i \leq 1$, für $1 \leq i \leq n$.

Gesucht:

Die kleinst mögliche Anzahl von Kisten (Bins) der Größe 1, mit der alle Objekte verpackt werden können.

Beispiel:

7 Objekte mit Größen 0.2, 0.5, 0.4, 0.7, 0.1, 0.3, 0.8

Online Bin Packing:

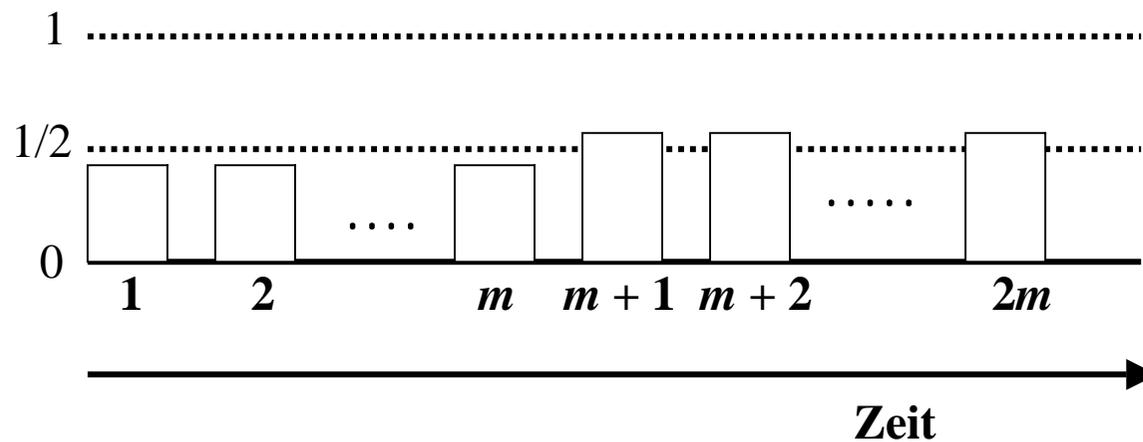
Jedes (ankommende) Objekt muss verpackt sein, bevor das nächste Objekt betrachtet wird. Ein Objekt verbleibt in derjenigen Kiste, in die es zuerst gepackt wird.

Offline Bin Packing:

Zunächst werden die Anzahl n und alle n Objekte vorgegeben. Dann beginnt die Verpackung.

Beobachtung

- Bin Packing ist beweisbar schwer.
(Offline Version ist NP-schwer.
Entscheidungsproblem ist NP-vollständig.)
- Kein Online Bin-Packing Verfahren kann stets eine optimale Lösung liefern



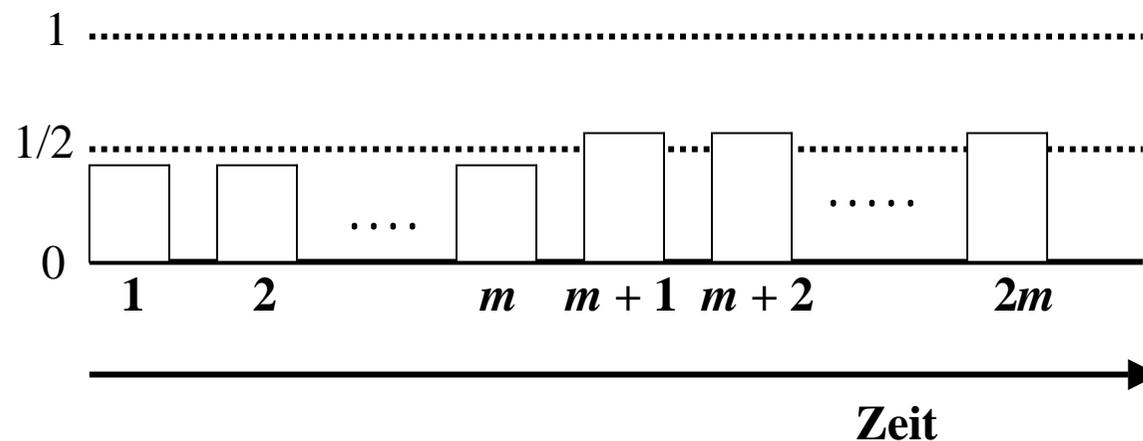
Online Verfahren

Satz 1

Es gibt Eingaben, die jeden Online Bin Packing Algorithmus zwingen, wenigstens $4/3$ OPT Bins zu verwenden, wobei OPT die minimal mögliche Binanzahl ist.

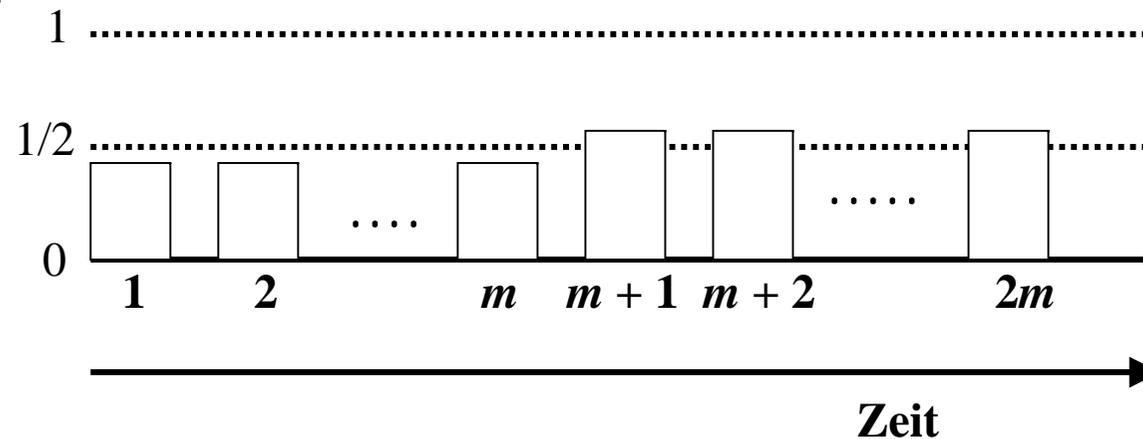
Beweis:

Annahme: Online Bin Packing Algorithmus A benötigt stets weniger als $4/3$ OPT Bins



Online Verfahren

Zeitpunkt 1:



$$OPT = m/2 \text{ und } \#Bins(A) = b$$

$$\text{Es gilt nach Annahme: } b < 4/3 \cdot m/2 = 2/3m$$

Sei $b_1 = \#Bins$ mit einem Objekt

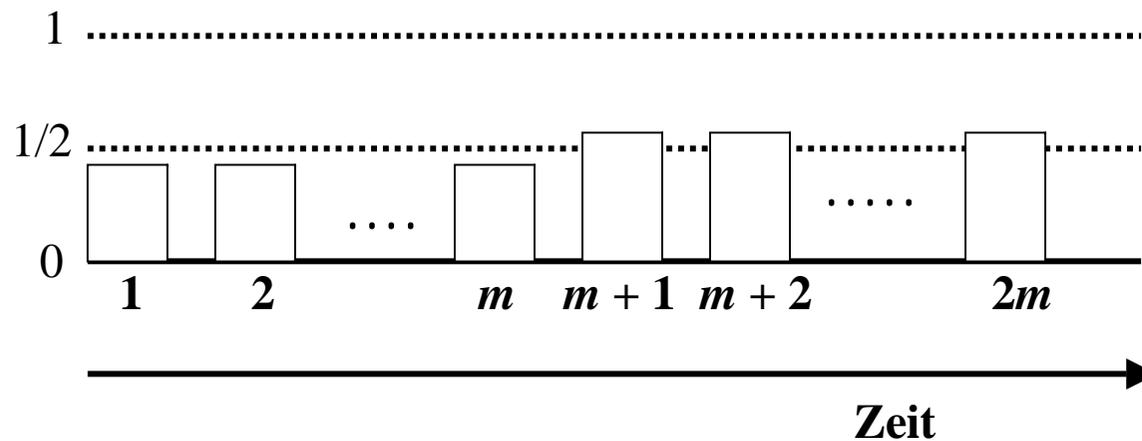
$b_2 = \#Bins$ mit zwei Objekten

$$\text{Es gilt: } b_1 + 2 b_2 = m$$

Online Verfahren



Zeitpunkt 2:



$$OPT = m \text{ und } \#Bins(A) \geq (b_1 + b_2) + (m - b_1) = m + b_2$$

$$\text{Annahme: } m + b_2 \leq \#Bins(A) < 4/3 m$$

$$b_2 < m/3$$

Next Fit (NF), First-Fit (FF), Best-Fit (BF)

Next Fit:

Verpacke das nächste Objekt in dieselbe Kiste wie das vorherige, wenn es dort noch hineinpasst, sonst öffne eine neue Kiste und verpacke es dort.

Satz 2

(a) Für alle Inputfolgen I gilt:

$$NF(I) \leq 2OPT(I).$$

(b) Es gibt Inputfolgen I mit:

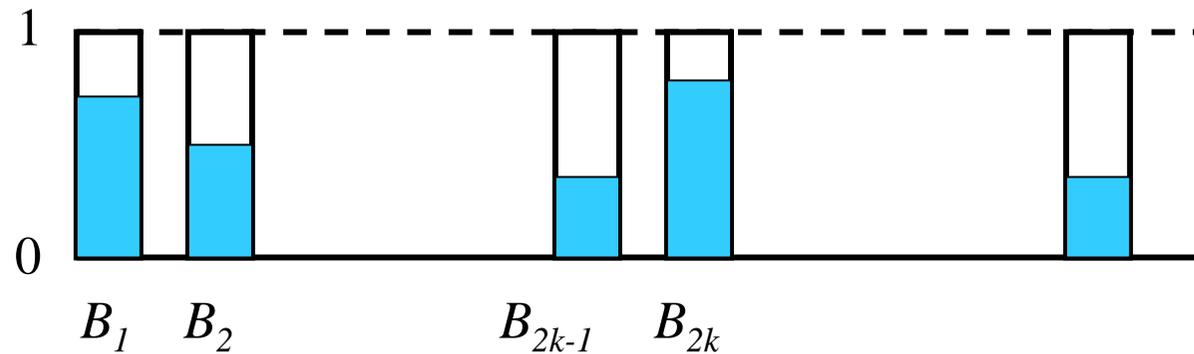
$$NF(I) \geq 2OPT(I) - 2.$$

Next Fit



Beweis: (a)

Betrachte zwei Kisten $B_{2k-1}, B_{2k}, 2k \leq NF(I)$.



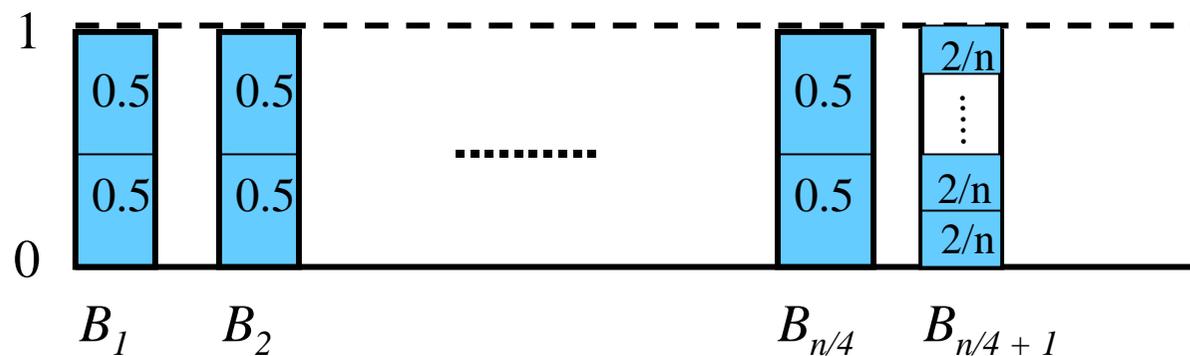
Next Fit

Beweis: (b)

Betrachte Inputfolge I mit Länge n
 ($n \equiv 0 \pmod{4}$):

$0.5, 2/n, 0.5, 2/n, 0.5, \dots, 0.5, 2/n$

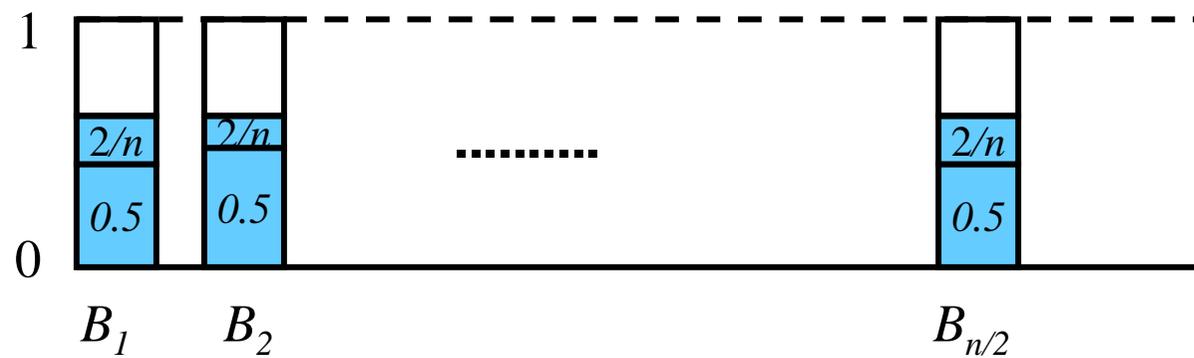
Optimale Packung:



Next Fit



Next Fit Liefert:



$$NF(I) =$$

$$OPT(I) =$$

First Fit:

Packe nächstes Objekt in die erste Kiste, in die es noch hineinpasst, wenn es eine solche Kiste gibt, sonst in eine neue Kiste.

Beobachtung:

Zu jedem Zeitpunkt kann es höchstens eine Kiste geben, die weniger als halb voll ist.

$$\rightarrow FF(I) \leq 2OPT(I)$$

Satz 3

(a) Für alle Inputfolgen I gilt:

$$FF(I) \leq \lceil 17/10 OPT(I) \rceil$$

(b) Es gibt Inputfolgen I mit:

$$FF(I) \geq 17/10 (OPT(I) - 1)$$

(b') Es gibt Inputfolgen I mit:

$$FF(I) = 10/6 OPT(I)$$

First Fit

Beweis (b`): Inputfolge der Länge $3 \cdot 6m$

$$\underbrace{1/7 + \varepsilon, \dots, 1/7 + \varepsilon}_{6m}, \underbrace{1/3 + \varepsilon, \dots, 1/3 + \varepsilon}_{6m},$$

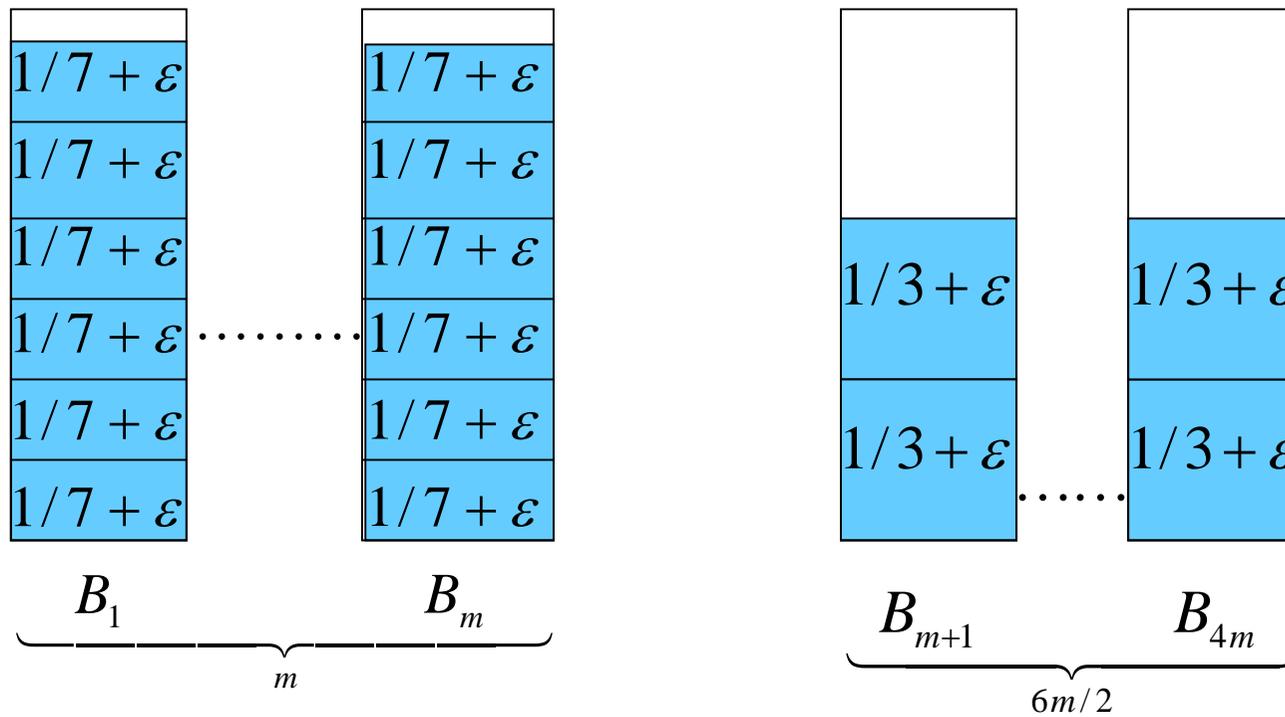
$$\underbrace{1/2 + \varepsilon, \dots, 1/2 + \varepsilon}_{6m}$$



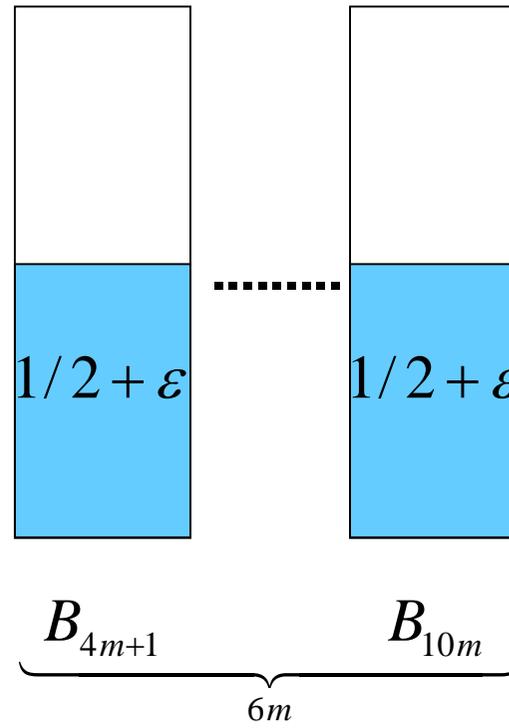
First Fit



First-Fit liefert:



First Fit



Best Fit:

Verpacke das nächste Objekt in diejenige Kiste, in die es am besten passt (d.h. den geringsten Platz ungenutzt lässt).

Verhalten von BF ähnlich zu FF

Laufzeit für Inputfolgen der Länge n

NF	$O(n)$		
FF	$O(n^2)$	→	$O(n \log n)$
BF	$O(n^2)$	→	$O(n \log n)$

Off-line Verfahren

n und s_1, \dots, s_n sind gegeben, bevor die Verpackung beginnt

Optimale Packung kann durch erschöpfende Suche bestimmt werden.

Idee für off-line Approximationsalgorithmus:

Sortiere die Objekte zunächst nach abnehmender Größe und verpacke größerer Objekte zuerst!

First Fit Decreasing (FFD) bzw. **FFNI**

Best Fit Decreasing (BFD)

First Fit Decreasing

Lemma 1

Sei I eine Folge von n Objekten mit Größen

$$s_1 \geq s_2 \geq \dots \geq s_n$$

und sei $m = OPT(I)$.

Dann haben alle von FFD in den Bins

$$B_{m+1}, B_{m+2}, \dots, B_{FFD(I)}$$

verpackten Objekte eine Größe von höchstens $1/3$.

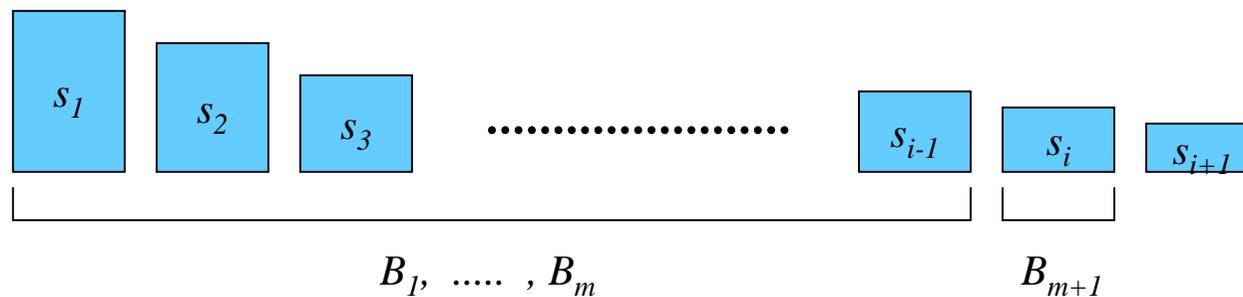
First Fit Decreasing

Beweis von Lemma 1:

Inputfolge I , absteigend sortiert

$m = OPT(I)$

i – kleinster Index, so dass $s_i \in B_{m+1}$



First Fit Decreasing

Beobachtung:

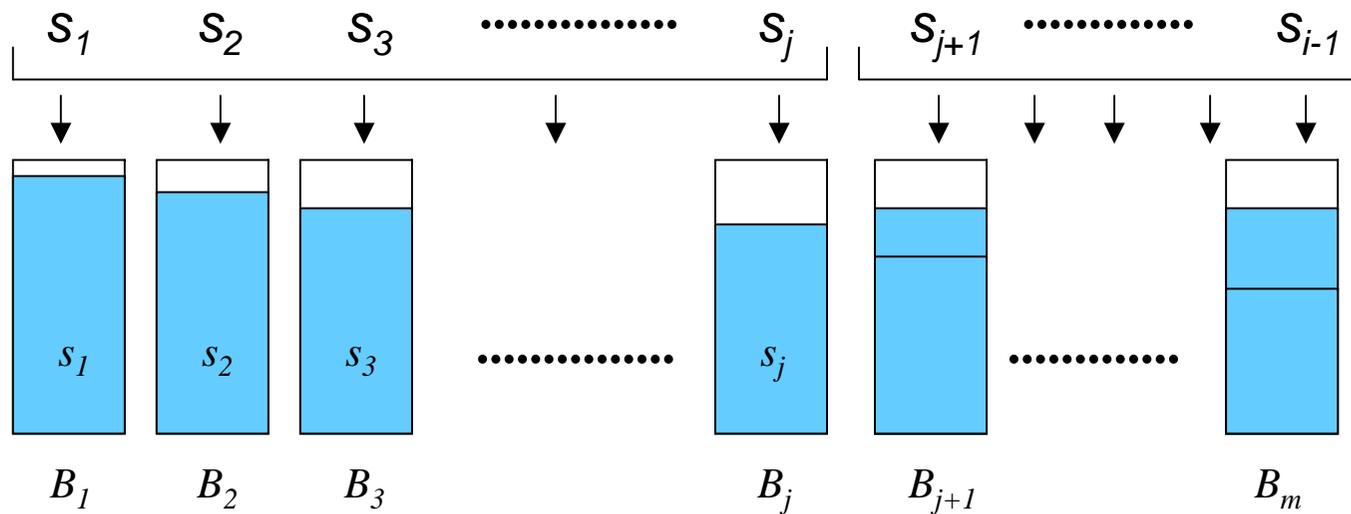
Gibt es eine Kiste mit drei Objekten in einer Verpackung von s_1, \dots, s_j , dann ist $s_j \leq 1/3$

Annahme:

Alle Kisten in der *FFD*-Verpackung von s_1, \dots, s_j enthalten höchstens zwei Objekte.

First Fit Decreasing

FFD-Verpackung



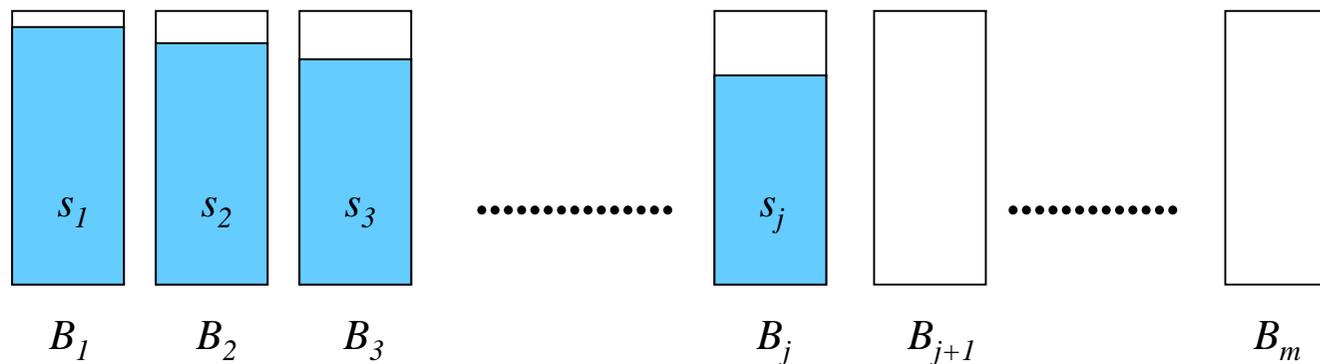
Gesamtanzahl der Objekte

$$i - 1 = j + 2(m - j)$$

First Fit Decreasing

Optimale Verpackung:

1. j Objekte s_1, \dots, s_j in j Kisten B_1, \dots, B_j
2. $i - j$ Objekte s_{j+1}, \dots, s_i in
 $m - j$ Kisten B_{j+1}, \dots, B_m



First Fit Decreasing

Lemma 2

Sei I eine Folge von n Objekten mit Größen

$$s_1 \geq s_2 \geq \dots \geq s_n$$

und sei $m = OPT(I)$.

Dann ist die Anzahl der Objekte, die *FFD* in die Kisten

$$B_{m+1}, B_{m+2}, \dots, B_{FFD(I)}$$

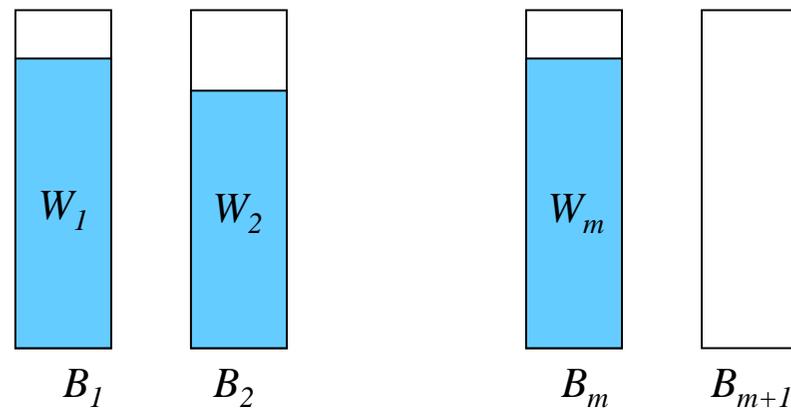
verpackt, höchstens $m - 1$.

First Fit Decreasing

Beweis von Lemma 2:

Annahme: Es gibt mehr als $m - 1$ Objekte

x_1, \dots, x_m in I , die FFD in extra Kisten verpackt.



First Fit Decreasing

Satz

Für alle Inputfolgen I gilt:

$$FFD(I) \leq (4 \cdot OPT(I) + 1)/3.$$

First Fit Decreasing

Satz

Für alle Inputfolgen I gilt:

$$FFD(I) \leq (4 \cdot OPT(I) + 1)/3.$$

Satz:

1. Für alle Inputfolgen I gilt:

$$FFD(I) \leq 11/9 \cdot OPT(I) + 4.$$

2. Es gibt Inputfolgen I mit:

$$FFD(I) = 11/9 \cdot OPT(I).$$

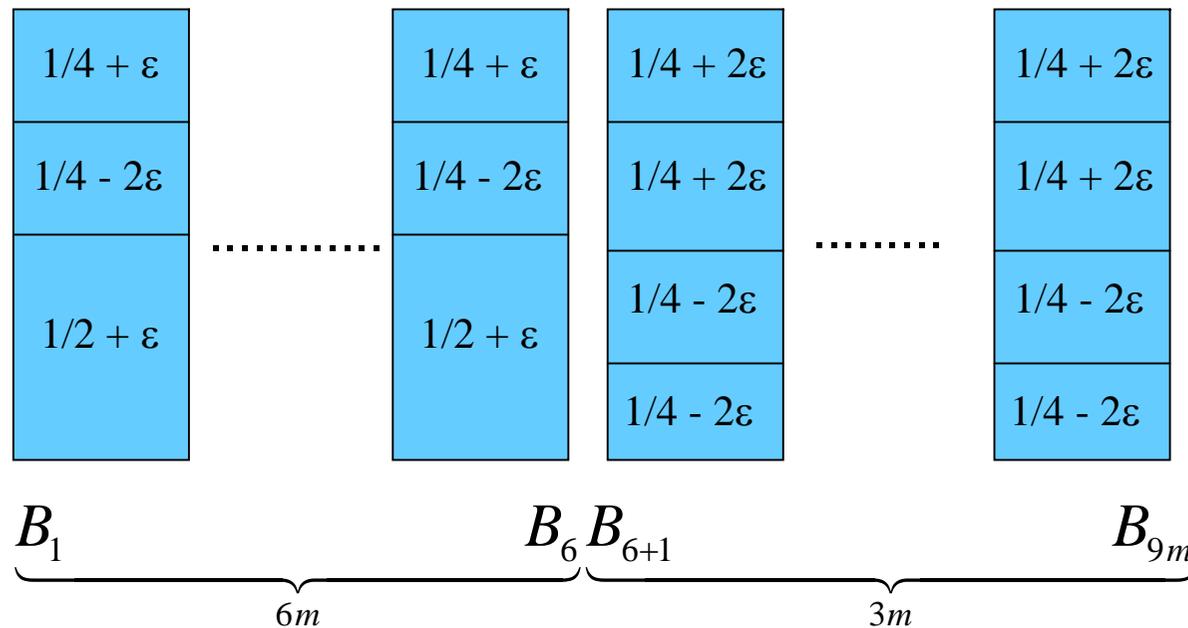
First Fit Decreasing

Beweis 2: Inputfolge der Länge $3 \cdot 6m + 12m$

$$\underbrace{1/2 + \varepsilon, \dots, 1/2 + \varepsilon}_{6m}, \underbrace{1/4 + 2\varepsilon, \dots, 1/4 + 2\varepsilon}_{6m}$$

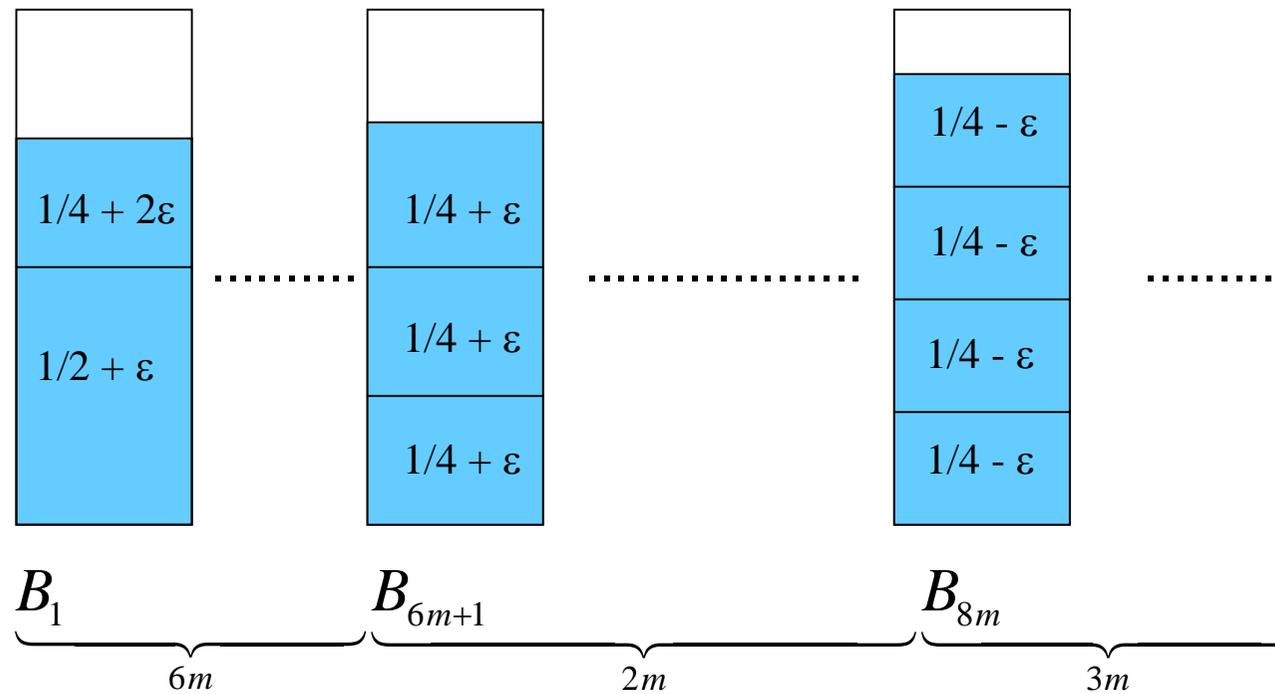
$$\underbrace{1/4 + \varepsilon, \dots, 1/4 + \varepsilon}_{6m}, \underbrace{1/4 - 2\varepsilon, \dots, 1/4 - 2\varepsilon}_{12m}$$

Optimale Packung:



First Fit Decreasing

First Fit Decreasing liefert:



$$OPT(I) = 9m$$

$$FFD(I) = 11m$$

Literaturhinweis

Bin Packing Verfahren sind Beispiele für Greedy Algorithmen
Werden in vielen Lehrbüchern behandelt, z.B. in:

Mark Allen Weiss: Data Structures and Algorithm Analysis in C++,
The Benjamin/Cummings Publishing Comp., Redwood City,
1994

Vgl. dort Chapter 10.1.3