Online Algorithms
Tutorial 1 — Bin Packing

Book chapter:

https://link.springer.com/content/pdf/10.1007%2FBFb0029568.pdf

Bin Packing

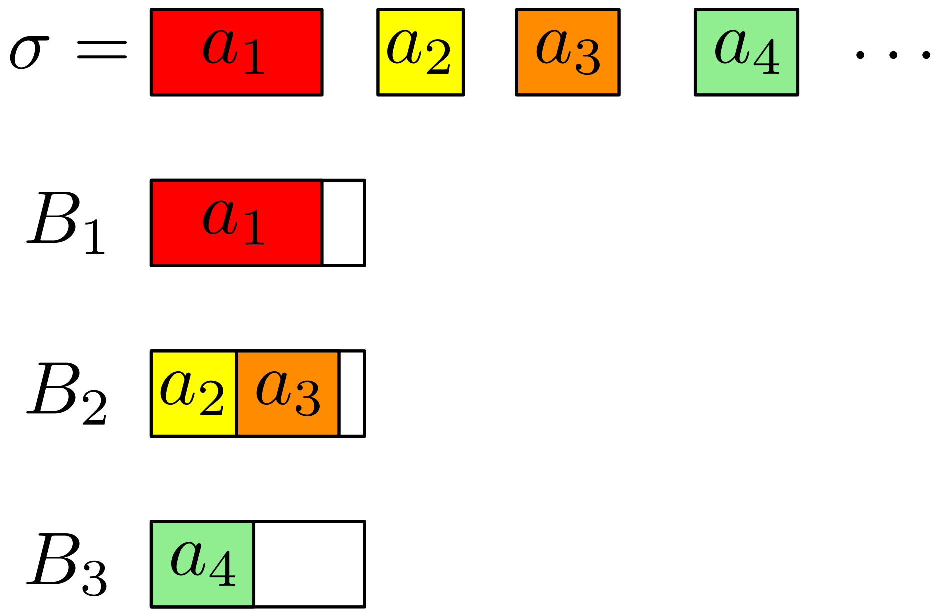
Bin Packing Problem:

- Item sequence $\sigma = a_1 a_2 \ldots$
- Item weights in $(0, 1]$
- Pack into bins with capacity 1
- Minimize number of bins
- Online: Pack a_i before knowing a_{i+1}

Bin Packing

Bin Packing Problem:

- Item sequence $\sigma = a_1 a_2 \ldots$
- Item weights in $(0, 1]$
- Pack into bins with capacity 1
- Minimize number of bins
- Online: Pack a_i before knowing a_{i+1}



Bin Packing

- NP-hard
  - Easy reduction: PARTITION or SUBSET SUM
  - Strong NP-hardness: 3-PARTITION
- Among the first online problems studied
- Adversary was first used here as argument by Yao
- Very thoroughly studied; we focus on classic results

Absolute Competitive Ratio

What we have used so far:

$$\text{(Absolute) competitive ratio: } \sup_{\sigma} \left\{ \frac{A(\sigma)}{\mathsf{OPT}(\sigma)} \right\}$$

Not great for bin packing:

- Not robust
- Corner cases, short sequences
- Relatively easy to get the best possible factor $\Rightarrow$ Exercise!

Asymptotic Competitive Ratio

More robust measure: *Asymptotic competitive ratio*

$$\limsup_{n\to\infty}\left(\sup\left\{\frac{A(\sigma)}{n}\middle|\mathsf{OPT}(\sigma)=n\right\}\right)$$

Asymptotic Competitive Ratio

More robust measure: *Asymptotic competitive ratio*

$$\limsup_{n\to\infty} \left(\sup \left\{ \frac{A(\sigma)}{n} \middle| \mathsf{OPT}(\sigma) = n \right\} \right)$$

For divergent series that do not diverge towards ∞.
Normally, $\lim\limits_{n\to\infty}$ is sufficient.

Asymptotic Competitive Ratio

More robust measure: *Asymptotic competitive ratio*

$$\limsup_{n\to\infty}\left(\sup\left\{\frac{A(\sigma)}{n}\,\middle|\,\mathsf{OPT}(\sigma)=n\right\}\right)$$

Select the worst sequence.

For divergent series that do not diverge towards ∞.
Normally, $\lim\limits_{n\to\infty}$ is sufficient.

Asymptotic Competitive Ratio — Example

PARITY

Input: A sequence of $\#$-symbols of unknown length n.

Output: After the first symbol, we answer $\rho = 0$ or $\rho = 1$.

Cost: $10 + f(n)$, where

$$f(n) := \begin{cases} n & \text{if } n \text{ is even}, \rho = 0, \\ n & \text{if } n \text{ is odd}, \rho = 1, \\ 2n & \text{if } n \text{ is odd}, \rho = 0, \\ 2n & \text{if } n \text{ is even}, \rho = 1. \end{cases}$$

Asymptotic Competitive Ratio — Example

Parity

Input: A sequence of #-symbols of unknown length n.

Output: After the first symbol, we answer $\rho = 0$ or $\rho = 1$.

Cost: $10 + f(n)$, where

$$f(n) := \begin{cases} n & \text{if } n \text{ is even}, \rho = 0, \\ n & \text{if } n \text{ is odd}, \rho = 1, \\ 2n & \text{if } n \text{ is odd}, \rho = 0, \\ 2n & \text{if } n \text{ is even}, \rho = 1. \end{cases}$$

Assume our algorithm guesses $\rho = 0$:

Asymptotic Competitive Ratio — Example

PARITY

Input: A sequence of #-symbols of unknown length n.

Output: After the first symbol, we answer $\rho = 0$ or $\rho = 1$.

Cost: $10 + f(n)$, where

$$f(n) := \begin{cases} n & \text{if } n \text{ is even}, \rho = 0, \\ n & \text{if } n \text{ is odd}, \rho = 1, \\ 2n & \text{if } n \text{ is odd}, \rho = 0, \\ 2n & \text{if } n \text{ is even}, \rho = 1. \end{cases}$$

Assume our algorithm guesses $\rho = 0$:

Even n

$\sup \frac{A}{\text{OPT}} = \frac{10+n}{10+n} = 1$

Asymptotic Competitive Ratio — Example

PARITY

Input: A sequence of $\#$-symbols of unknown length n.

Output: After the first symbol, we answer $\rho = 0$ or $\rho = 1$.

Cost: $10 + f(n)$, where

$$f(n) := \begin{cases} n & \text{if } n \text{ is even}, \rho = 0, \\ n & \text{if } n \text{ is odd}, \rho = 1, \\ 2n & \text{if } n \text{ is odd}, \rho = 0, \\ 2n & \text{if } n \text{ is even}, \rho = 1. \end{cases}$$

Assume our algorithm guesses $\rho = 0$:

Even n

$\sup \frac{A}{\text{OPT}} = \frac{10+n}{10+n} = 1$

Odd n

Asymptotic Competitive Ratio — Example

PARITY

Input: A sequence of #-symbols of unknown length n.

Output: After the first symbol, we answer $\rho = 0$ or $\rho = 1$.

Cost: $10 + f(n)$, where

$$f(n) := \begin{cases} n & \text{if } n \text{ is even}, \rho = 0, \\ n & \text{if } n \text{ is odd}, \rho = 1, \\ 2n & \text{if } n \text{ is odd}, \rho = 0, \\ 2n & \text{if } n \text{ is even}, \rho = 1. \end{cases}$$

Assume our algorithm guesses $\rho = 0$:

Even n

$\sup \frac{A}{\text{OPT}} = \frac{10+n}{10+n} = 1$

Odd n

$\sup \frac{A}{\text{OPT}} = \frac{10+2n}{10+n} = 2 - \varepsilon$

Asymptotic Competitive Ratio — Example

PARITY

Input: A sequence of #-symbols of unknown length n.

Output: After the first symbol, we answer $\rho = 0$ or $\rho = 1$.

Cost: $10 + f(n)$, where

$$f(n) := \begin{cases} n & \text{if } n \text{ is even}, \rho = 0, \\ n & \text{if } n \text{ is odd}, \rho = 1, \\ 2n & \text{if } n \text{ is odd}, \rho = 0, \\ 2n & \text{if } n \text{ is even}, \rho = 1. \end{cases}$$

Assume our algorithm guesses $\rho = 0$:

Even n

$\sup \frac{A}{\text{OPT}} = \frac{10+n}{10+n} = 1$

Odd n

$\sup \frac{A}{\text{OPT}} = \frac{10+2n}{10+n} = 2 - \varepsilon$

Asymptotic competitive ratio: 2

Asymptotic Competitive Ratio — Bin Packing

- Intuition: Small bad instances are ignored.
- Lower bound examples must generalize to arbitrary weight.
- Ignore constantly many bins: $\frac{A+k}{\text{OPT}}, \text{OPT} \to \infty$

Next Fit

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

NEXT FIT

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

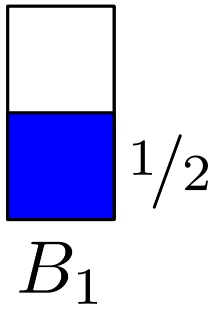
$$\sigma = {}^1/2, \varepsilon, {}^1/2, \varepsilon, \ldots$$

Online Bin Packing — Next Fit

Next Fit

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

$$\sigma = {}^1\!/_2, \varepsilon, {}^1\!/_2, \varepsilon, \ldots$$

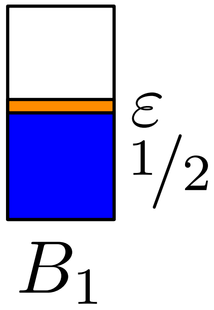


Online Bin Packing — Next Fit

Next Fit

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

$$\sigma = 1/2, \varepsilon, 1/2, \varepsilon, \ldots$$

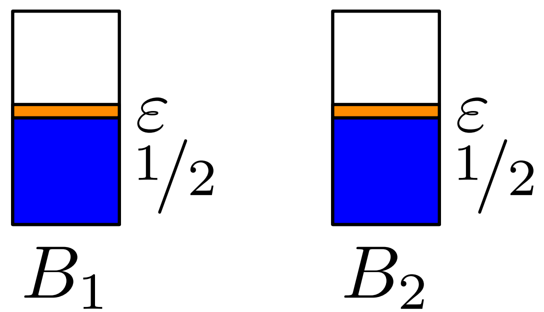


Online Bin Packing — Next Fit

NEXT FIT

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

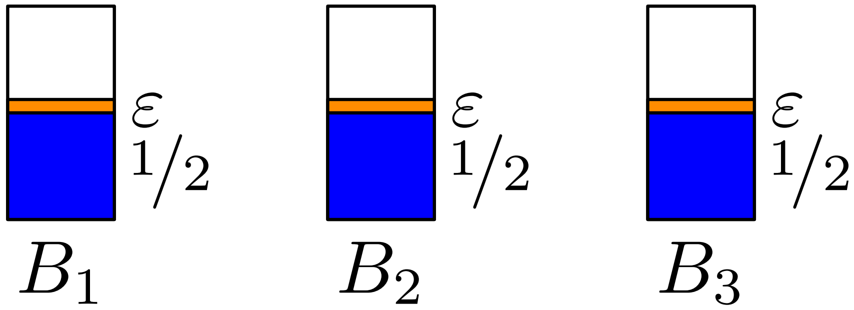
$$\sigma = {}^1\!/_2, \varepsilon, {}^1\!/_2, \varepsilon, \ldots$$



Next Fit

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

$$\sigma = 1/2, \varepsilon, 1/2, \varepsilon, \ldots$$

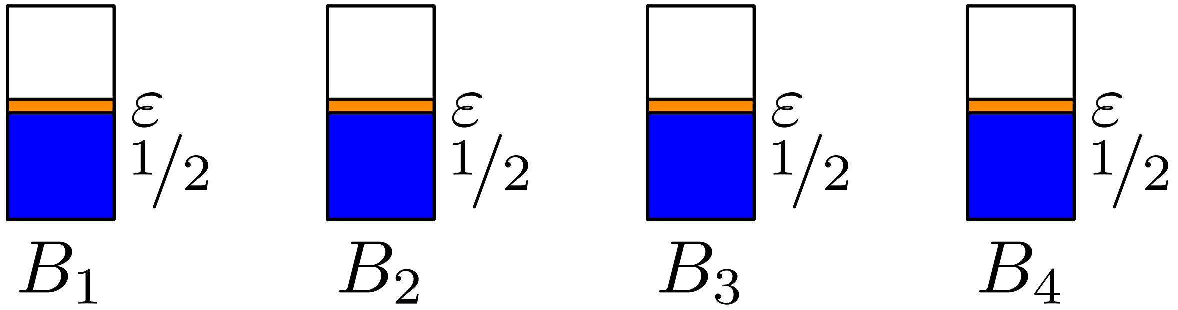


Online Bin Packing — Next Fit

Next Fit

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

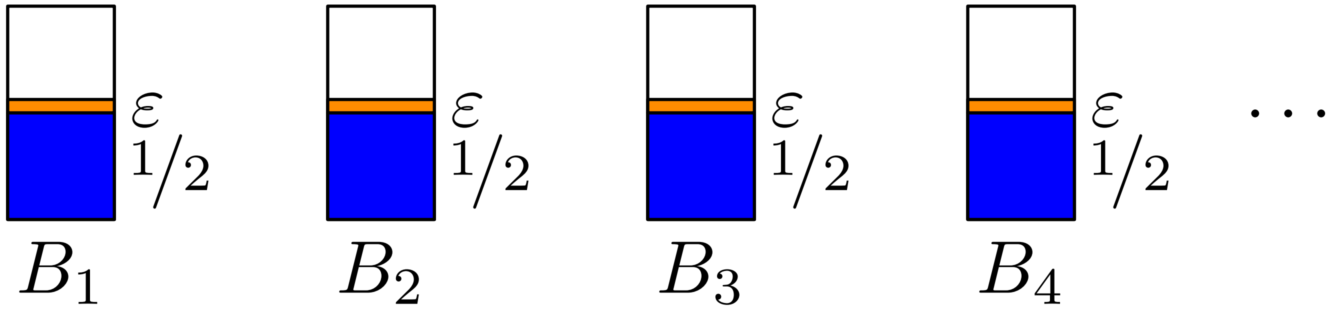
$$\sigma = {}^1\!/_2, \varepsilon, {}^1\!/_2, \varepsilon, \ldots$$



Next Fit

- Only one *active* bin B.
- If item a_i still fits into B, pack it there.
- Otherwise, close B and create a new bin B.
- Competitive ratio?

$$\sigma = {}^1\!/_2, \varepsilon, {}^1\!/_2, \varepsilon, \ldots$$



$$c \geq 2$$

Bin Packing — Next Fit

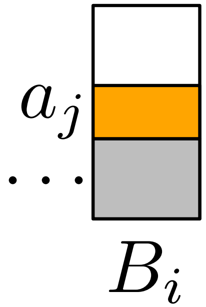
Can c be worse than 2?

- Consider B_i, B_{i+1}
- Why did we close B_i?

Bin Packing — Next Fit

Can c be worse than 2?

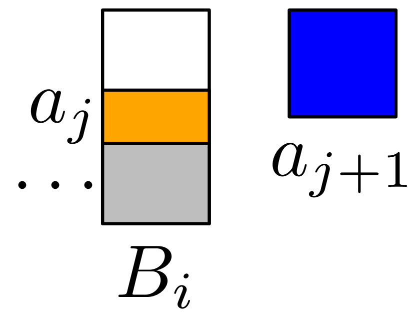
- Consider B_i, B_{i+1}
- Why did we close B_i?



Bin Packing — Next Fit

Can c be worse than 2?

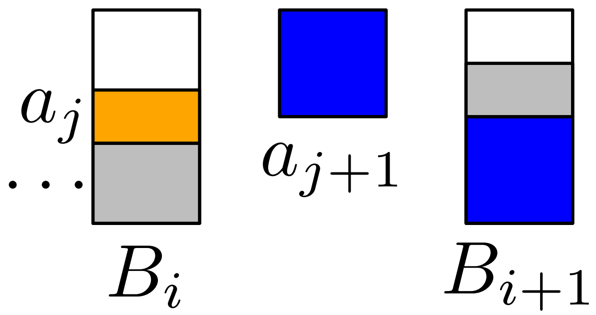
- Consider B_i, B_{i+1}
- Why did we close B_i?



Bin Packing — Next Fit

Can c be worse than 2?

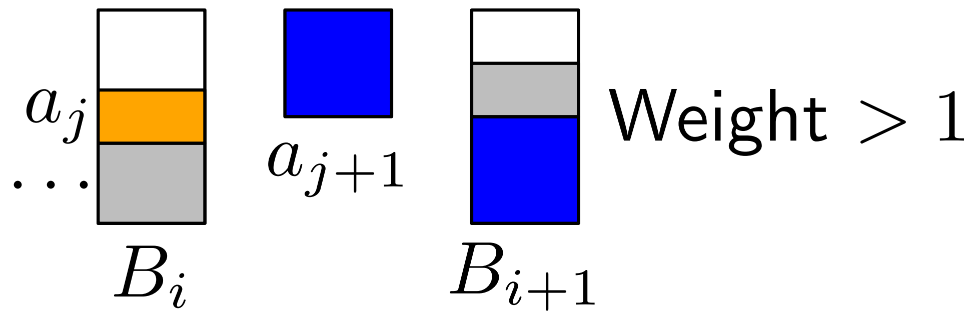
- Consider B_i, B_{i+1}
- Why did we close B_i?



Bin Packing — Next Fit

Can c be worse than 2?

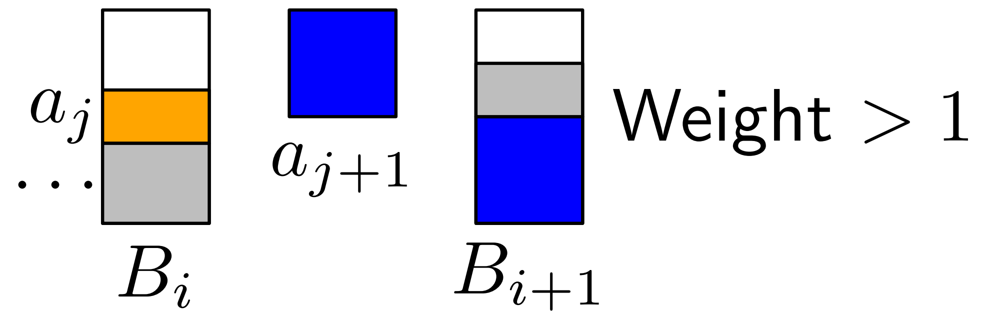
- Consider B_i, B_{i+1}
- Why did we close B_i?



Bin Packing — Next Fit

Can c be worse than 2?

- Consider B_i, B_{i+1}
- Why did we close B_i?

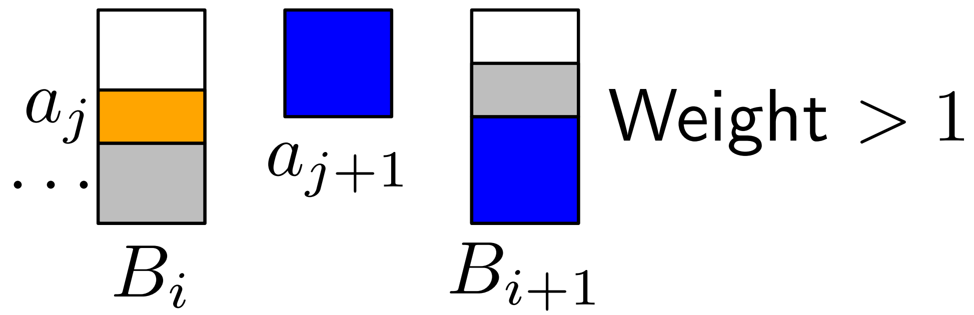


$\Rightarrow$ The items packed into B_i and B_{i+1} have weight at least 1!

Bin Packing — Next Fit

Can c be worse than 2?

- Consider B_i, B_{i+1}
- Why did we close B_i?



$\Rightarrow$ The items packed into B_i and B_{i+1} have weight at least 1!

Let b be our number of bins.

- OPT cannot pack more than weight 1 per bin.

$\Rightarrow$ $b \leq 2 \cdot \text{OPT} + 1$

- Asymptotic competitive ratio: 2
- General lower bound? Better algorithms?

Bin Packing — First Fit

Simplest extension: First Fit

- All our bins are left open
- If the item fits into any bin, just pick the first to pack
- Only open new bin if no bin fits

Bin Packing — First Fit

Simplest extension: First Fit

- All our bins are left open
- If the item fits into any bin, just pick the first to pack
- Only open new bin if no bin fits

How well does this handle $1/2, \varepsilon, 1/2, \varepsilon, \ldots$?

Bin Packing — First Fit

Simplest extension: First Fit

- All our bins are left open
- If the item fits into any bin, just pick the first to pack
- Only open new bin if no bin fits

How well does this handle $1/2, \varepsilon, 1/2, \varepsilon, \ldots$?

So, how do we trick this algorithm?

Bin Packing — First Fit

Simplest extension: First Fit

- All our bins are left open
- If the item fits into any bin, just pick the first to pack
- Only open new bin if no bin fits

How well does this handle $1/2, \varepsilon, 1/2, \varepsilon, \ldots$?

So, how do we trick this algorithm?

- Send in group of small items
- If they are packed together, send bigger
- Otherwise, end input
- Repeat in phases

Bin Packing — First Fit Bound

Three categories: $\underbrace{\frac{1}{6} - 2\varepsilon}_{\text{I: } n/3}$, $\underbrace{\frac{1}{3} + \varepsilon}_{\text{II: } n/3}$, $\underbrace{\frac{1}{2} + \varepsilon}_{\text{III: } n/3}$, n divisible by 18

Bin Packing — First Fit Bound

Three categories: $\underbrace{\frac{1}{6} - 2\varepsilon}_{\text{I: } n/3}$, $\underbrace{\frac{1}{3} + \varepsilon}_{\text{II: } n/3}$, $\underbrace{\frac{1}{2} + \varepsilon}_{\text{III: } n/3}$, n divisible by 18

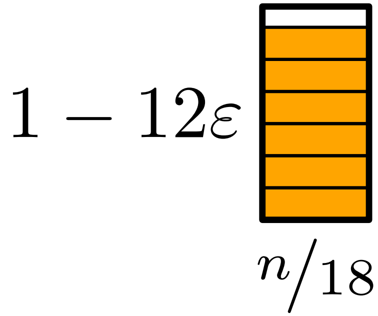
What does First Fit do?

Bin Packing — First Fit Bound

Three categories: $\frac{1}{6} - 2\varepsilon$, $\frac{1}{3} + \varepsilon$, $\frac{1}{2} + \varepsilon$, n divisible by 18

I: $n/3$ II: $n/3$ III: $n/3$

What does First Fit do?

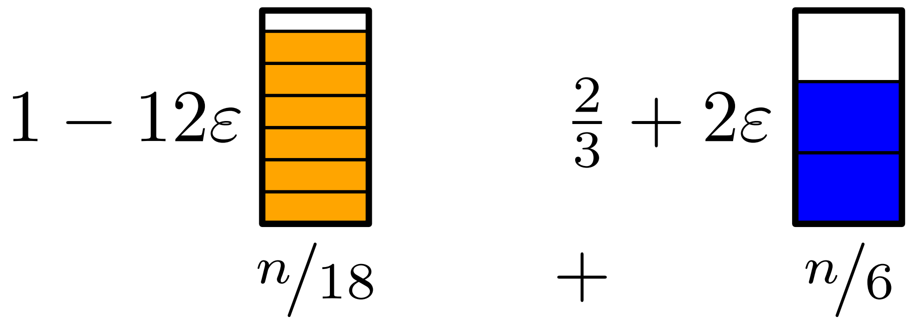


Bin Packing — First Fit Bound

Three categories: $\frac{1}{6} - 2\varepsilon$, $\frac{1}{3} + \varepsilon$, $\frac{1}{2} + \varepsilon$, n divisible by 18

I: $n/3$ II: $n/3$ III: $n/3$

What does FIRST FIT do?

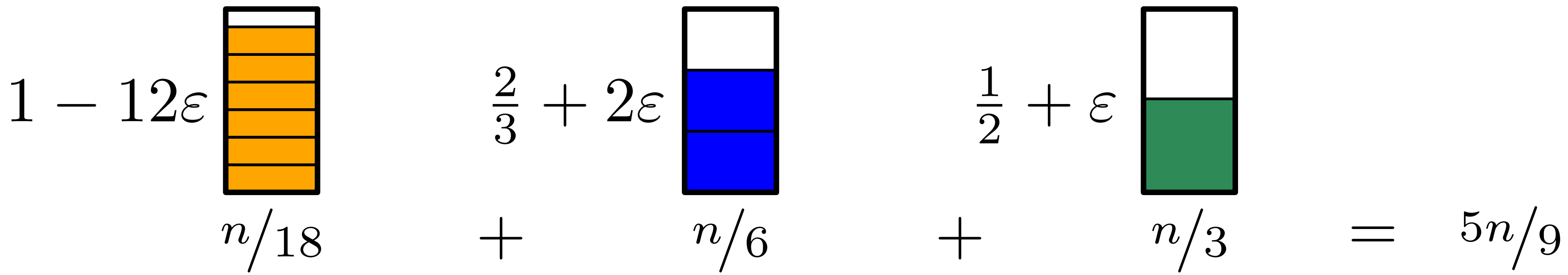


Bin Packing — First Fit Bound

Three categories: $\frac{1}{6} - 2\varepsilon$, $\frac{1}{3} + \varepsilon$, $\frac{1}{2} + \varepsilon$, n divisible by 18

I: $n/3$ II: $n/3$ III: $n/3$

What does First Fit do?

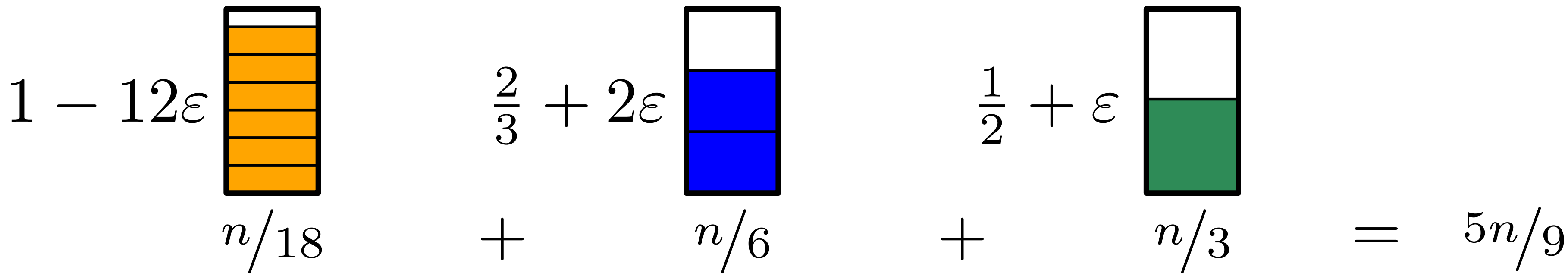


Bin Packing — First Fit Bound

Three categories: $\frac{1}{6} - 2\varepsilon$, $\frac{1}{3} + \varepsilon$, $\frac{1}{2} + \varepsilon$, n divisible by 18

I: $n/3$ II: $n/3$ III: $n/3$

What does FIRST FIT do?



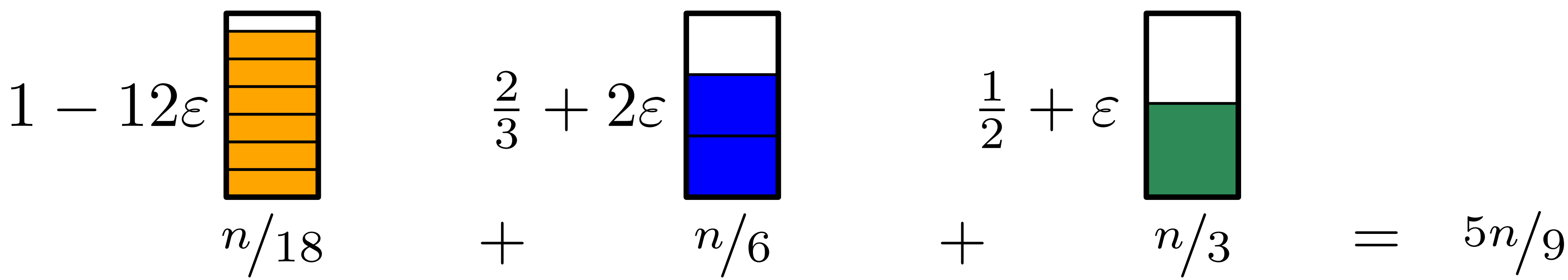
But what would OPT do?

Bin Packing — First Fit Bound

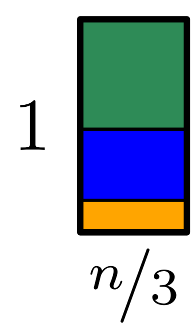
Three categories: $\frac{1}{6} - 2\varepsilon$, $\frac{1}{3} + \varepsilon$, $\frac{1}{2} + \varepsilon$, n divisible by 18

I: $n/3$ II: $n/3$ III: $n/3$

What does First Fit do?



But what would OPT do?

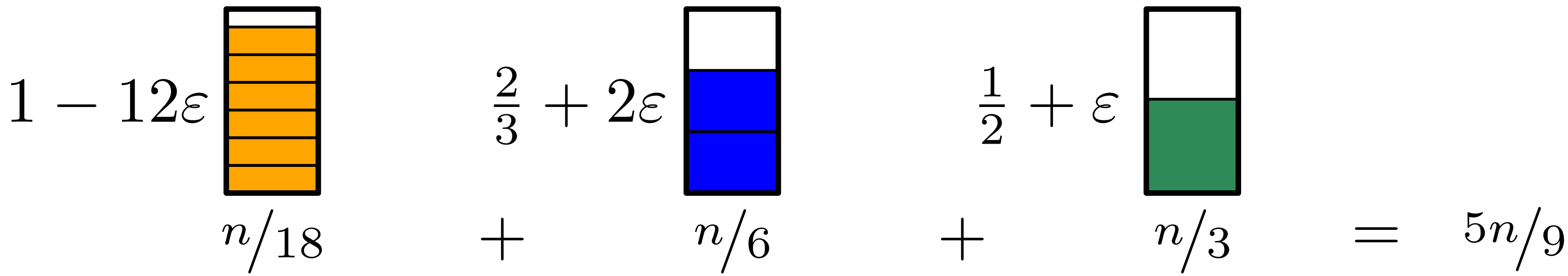


Bin Packing — First Fit Bound

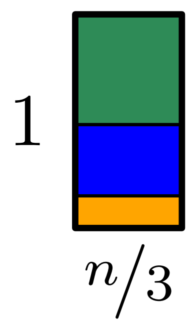
Three categories: $\frac{1}{6} - 2\varepsilon$, $\frac{1}{3} + \varepsilon$, $\frac{1}{2} + \varepsilon$, n divisible by 18

I: $n/3$ II: $n/3$ III: $n/3$

What does First Fit do?



But what would OPT do?



$$\Rightarrow c \geq \frac{5n/9}{n/3} = \frac{5}{3} \approx 1.666.$$

First Fit Bounds — Continued

Is that it? Can it be even worse?

First Fit Bounds — Continued

Is that it? Can it be even worse?

WORST-CASE PERFORMANCE BOUNDS FOR SIMPLE ONE-DIMENSIONAL PACKING ALGORITHMS*

D. S. JOHNSON†, A. DEMERS‡, J. D. ULLMAN§, M. R. GAREY‖ AND R. L. GRAHAM‖

Let N be a positive integer divisible by 17 and let δ be chosen so that $0 < \delta \ll 18^{-N/17}$. The first region will consist of $N/17$ *blocks* of ten numbers each. Let the numbers of the ith block of region 1 be denoted by $a_{0i}, a_{1i}, \cdots, a_{9i}$. These numbers are given by the following expressions, where $\delta_i = \delta \cdot 18^{(N/17)-i}$ for $1 \leqq i \leqq N/17$:

$$a_{0i} = \tfrac{1}{6} + 33\delta_i, \qquad a_{4i} = \tfrac{1}{6} - 13\delta_i,$$
$$a_{1i} = \tfrac{1}{6} - 3\delta_i, \qquad a_{5i} = \tfrac{1}{6} + 9\delta_i,$$
$$a_{2i} = a_{3i} = \tfrac{1}{6} - 7\delta_i, \qquad a_{6i} = a_{7i} = a_{8i} = a_{9i} = \tfrac{1}{6} - 2\delta_i.$$

Let the first $10N/17$ numbers in the list L be $a_{01}, a_{11}, \cdots, a_{91}, a_{02}, \cdots, a_{92},$

$$b_{0i} = \tfrac{1}{3} + 46\delta_i, \qquad b_{4i} = \tfrac{1}{3} + 12\delta_i,$$
$$b_{1i} = \tfrac{1}{3} - 34\delta_i, \qquad b_{5i} = \tfrac{1}{3} - 10\delta_i,$$
$$b_{2i} = b_{3i} = \tfrac{1}{3} + 6\delta_i, \qquad b_{6i} = b_{7i} = b_{8i} = b_{9i} = \tfrac{1}{3} + \delta_i.$$

Similar argument, more complicated input $\Rightarrow c \geq 1.7$

First Fit Bounds — Continued

Is that it? Can it be even worse?

WORST-CASE PERFORMANCE BOUNDS FOR SIMPLE ONE-DIMENSIONAL PACKING ALGORITHMS*

D. S. JOHNSON†, A. DEMERS‡, J. D. ULLMAN§, M. R. GAREY‖ AND R. L. GRAHAM‖

Let N be a positive integer divisible by 17 and let δ be chosen so that $0 < \delta \ll 18^{-N/17}$. The first region will consist of $N/17$ *blocks* of ten numbers each. Let the numbers of the ith block of region 1 be denoted by $a_{0i}, a_{1i}, \cdots, a_{9i}$. These numbers are given by the following expressions, where $\delta_i = \delta \cdot 18^{(N/17)-i}$ for $1 \leqq i \leqq N/17$:

$$a_{0i} = \tfrac{1}{6} + 33\delta_i, \qquad a_{4i} = \tfrac{1}{6} - 13\delta_i,$$
$$a_{1i} = \tfrac{1}{6} - 3\delta_i, \qquad a_{5i} = \tfrac{1}{6} + 9\delta_i,$$
$$a_{2i} = a_{3i} = \tfrac{1}{6} - 7\delta_i, \qquad a_{6i} = a_{7i} = a_{8i} = a_{9i} = \tfrac{1}{6} - 2\delta_i.$$

Let the first $10N/17$ numbers in the list L be $a_{01}, a_{11}, \cdots, a_{91}, a_{02}, \cdots, a_{92},$

$$b_{0i} = \tfrac{1}{3} + 46\delta_i, \qquad b_{4i} = \tfrac{1}{3} + 12\delta_i,$$
$$b_{1i} = \tfrac{1}{3} - 34\delta_i, \qquad b_{5i} = \tfrac{1}{3} - 10\delta_i,$$
$$b_{2i} = b_{3i} = \tfrac{1}{3} + 6\delta_i, \qquad b_{6i} = b_{7i} = b_{8i} = b_{9i} = \tfrac{1}{3} + \delta_i.$$

Similar argument, more complicated input $\Rightarrow c \geq 1.7$
Works for all ANY FIT algorithms!

First Fit Upper Bound

Is FIRST FIT 1.7-competitive?

Yes! Proof idea:

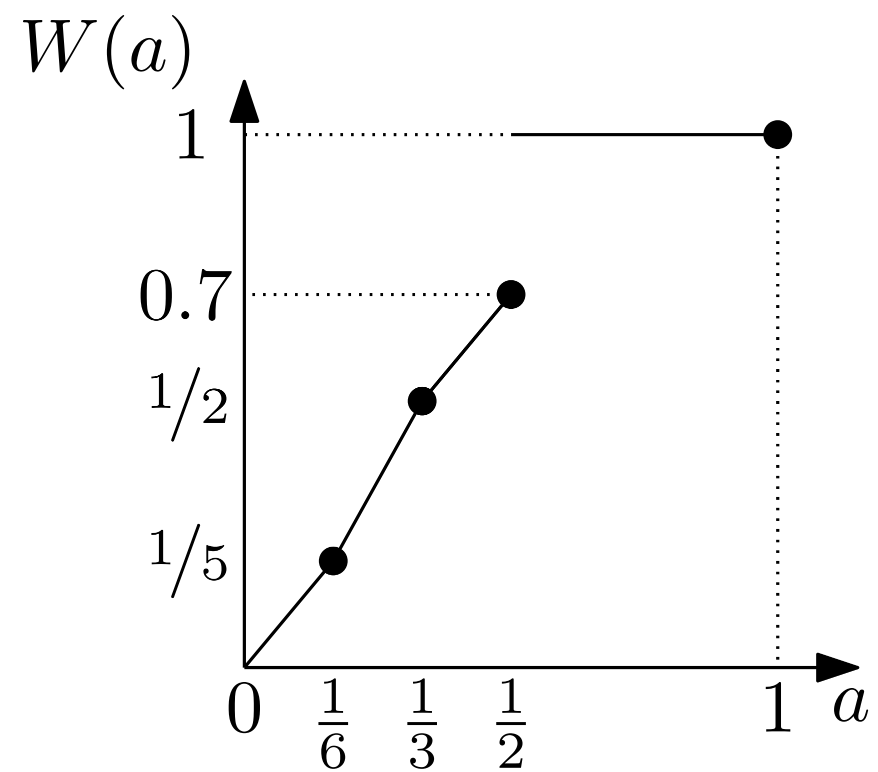
- Weight function $W : [0, 1] \to [0, 1]$
- Maps item weight a_i to $W(a_i) \geq a_i$
- $W(a)$ bounds how much space we may need to pack a
- We prove FF$(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2 \leq 1.7OPT(\sigma) + 2$.

First Fit Upper Bound

Is First Fit 1.7-competitive?
Yes! Proof idea:

- Weight function $W : [0,1] \to [0,1]$
- Maps item weight a_i to $W(a_i) \geq a_i$
- $W(a)$ bounds how much space we may need to pack a
- We prove $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2 \leq 1.7\mathrm{OPT}(\sigma) + 2$.

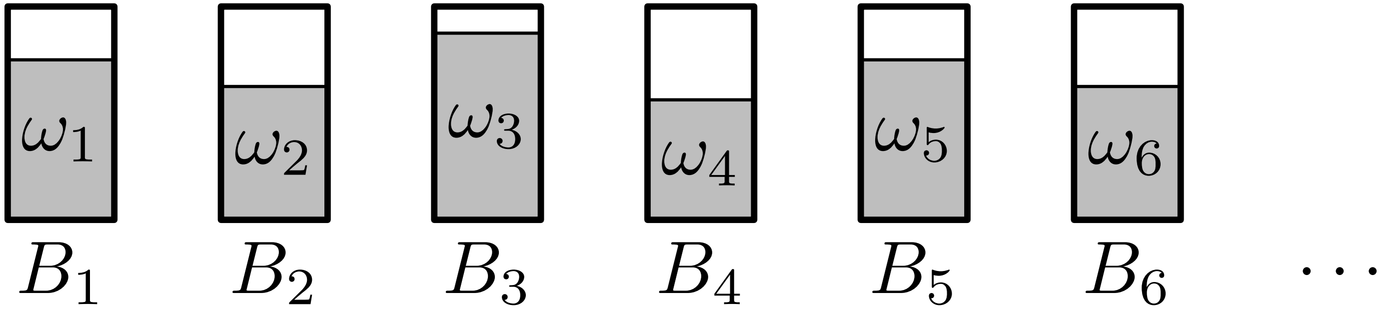
$$W(a) = \begin{cases} 6a/5 & \text{if } a \leq 1/6 \\ 9a/5 - 1/10 & \text{else if } a \leq 1/3 \\ 6a/5 + 1/10 & \text{else if } a \leq 1/2 \\ 1 & 1/2 < a \end{cases}$$



First Fit — Coarseness

Coarseness α_i of a bin B_i: Smallest item in B_i

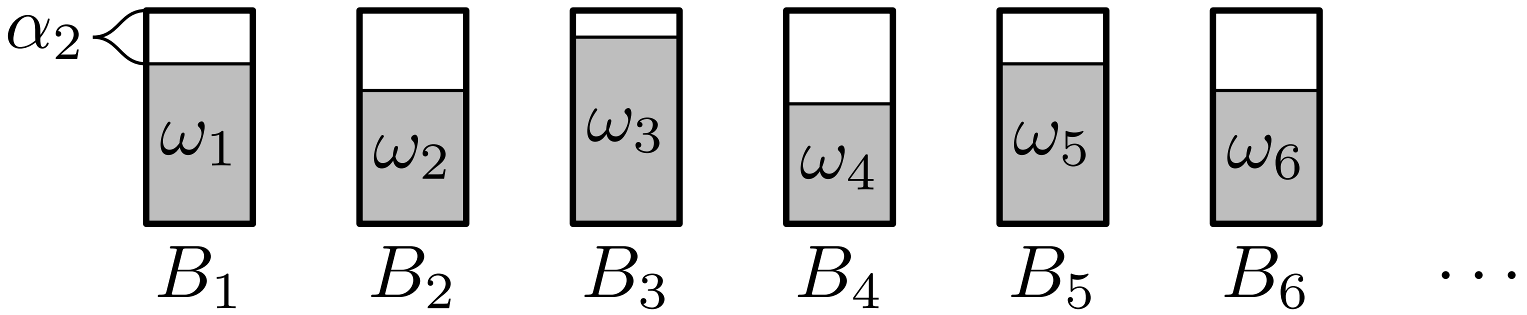
$$\alpha_i < 1 - \min_{j<i} \omega_j$$



First Fit — Coarseness

Coarseness α_i of a bin B_i: Smallest item in B_i

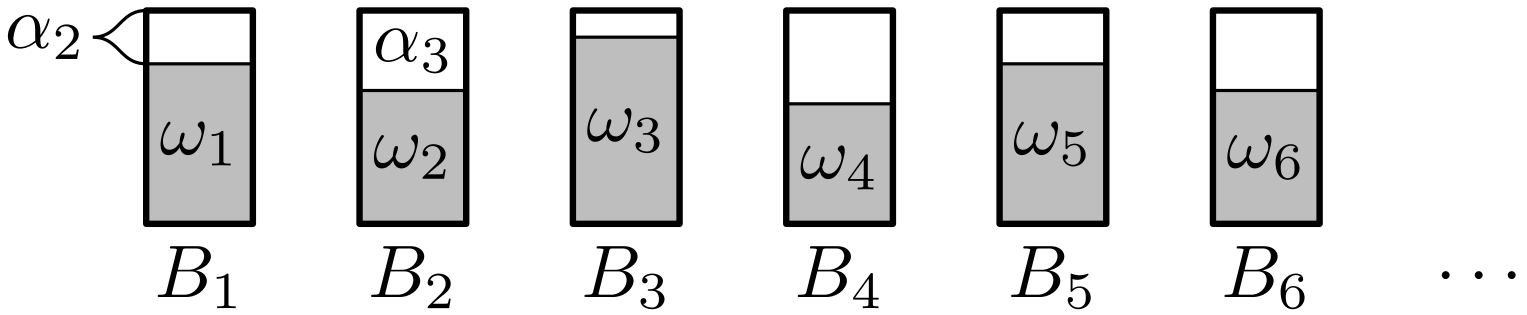
$$\alpha_i < 1 - \min_{j<i} \omega_j$$



First Fit — Coarseness

Coarseness α_i of a bin B_i: Smallest item in B_i

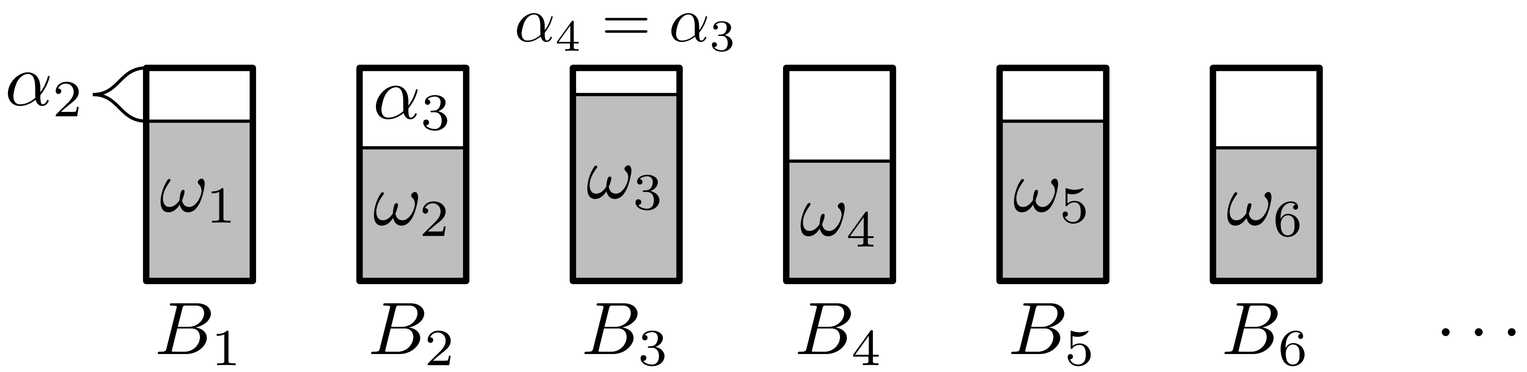
$$\alpha_i < 1 - \min_{j<i} \omega_j$$



First Fit — Coarseness

Coarseness α_i of a bin B_i: Smallest item in B_i

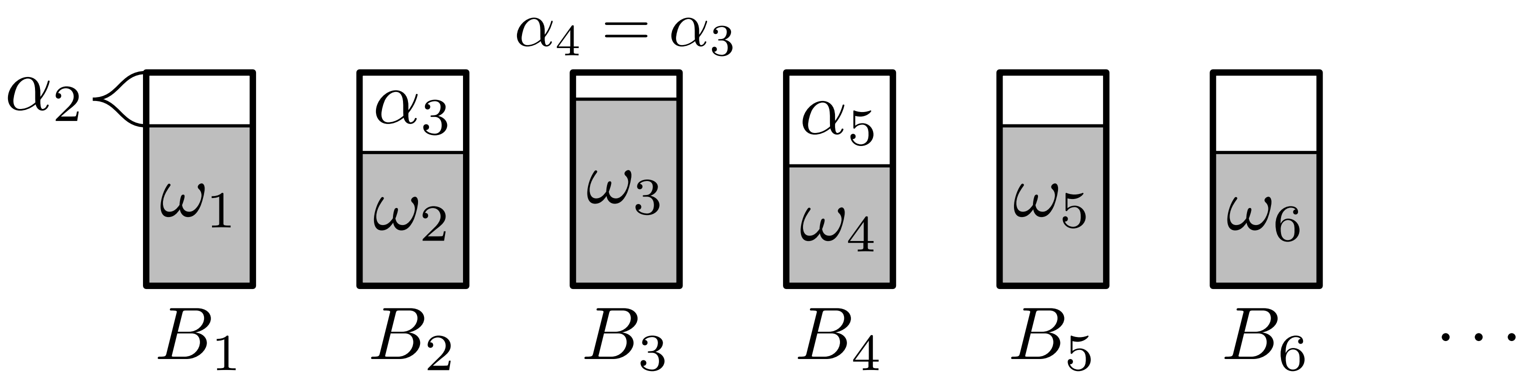
$$\alpha_i < 1 - \min_{j<i} \omega_j$$



First Fit — Coarseness

Coarseness α_i of a bin B_i: Smallest item in B_i

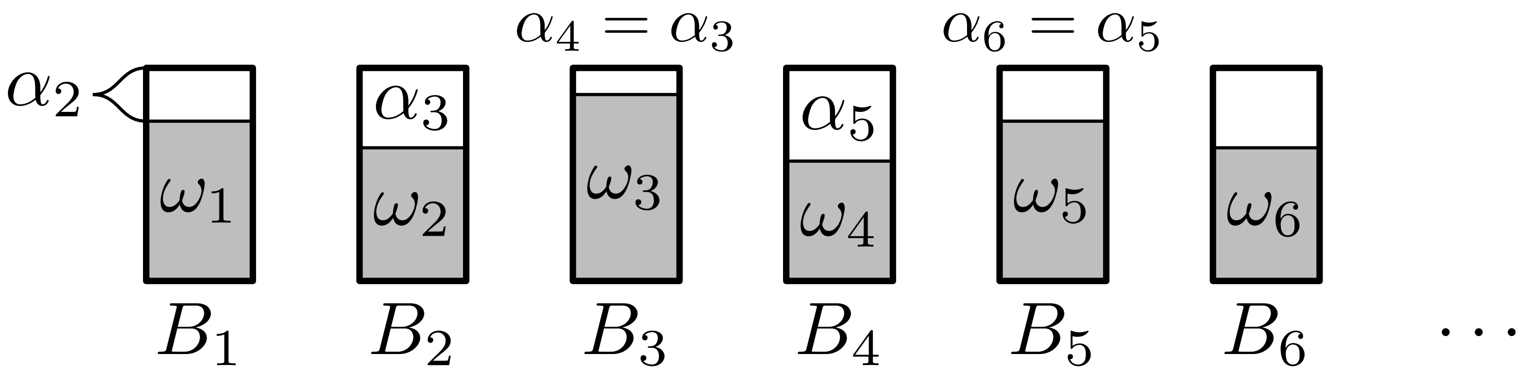
$$\alpha_i < 1 - \min_{j<i} \omega_j$$



First Fit — Coarseness

Coarseness α_i of a bin B_i: Smallest item in B_i

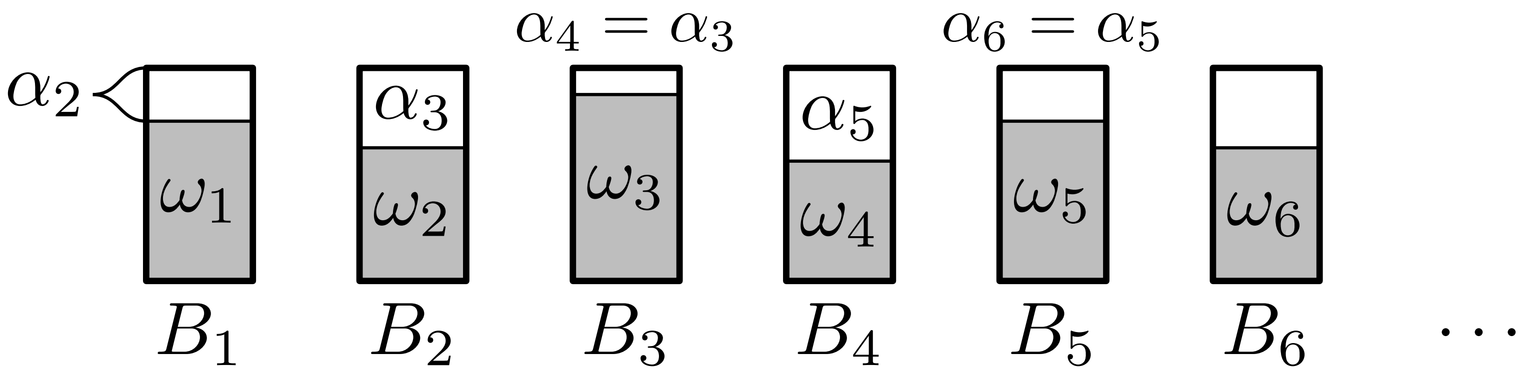
$$\alpha_i < 1 - \min_{j<i} \omega_j$$



First Fit — Coarseness

Coarseness α_i of a bin B_i: Smallest item in B_i

$$\alpha_i < 1 - \min_{j<i} \omega_j$$



Idea:

- Any item placed in B_i must exceed α_i

First Fit — First step

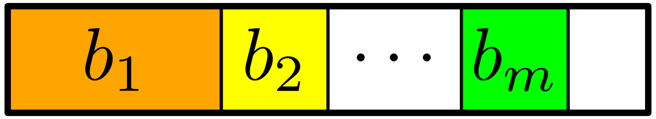
Proof step 1: $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

First Fit — First step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 1: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$

B_i

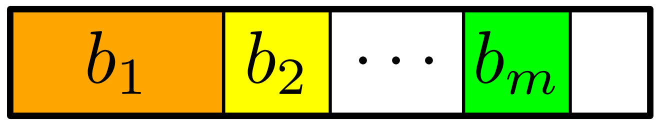


First Fit — First step

Proof step 1: $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 1: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$

B_i



If $\sum_{\ell=1}^{m} b_\ell > 1 - \alpha$, then $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$.

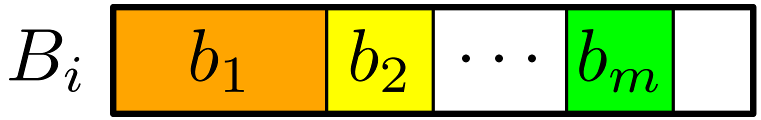
Proof:

- $b_1 > 1/2 \Rightarrow W(b_1) = 1$, so we assume $b_1 \leq 1/2$
- So at least $b_1 \geq b_2 > \alpha$
- Case distinction on α:
  - $\alpha \leq 1/6$: $W(\beta)/\beta \geq \frac{6}{5}$ for $\beta \leq 1/2$, so $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$.
  - $\alpha \geq 1/3$: $b_1 \geq b_2 \geq 1/3 \Rightarrow W(b_1) + W(b_2) \geq 1$.
  - The case $\alpha \in (1/6, 1/3)$ needs subcases on m

First Fit — First Step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 1: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$



If $\sum_{\ell=1}^{m} b_\ell > 1 - \alpha$, then $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$.

Proof for $\alpha \in (1/6, 1/3)$:

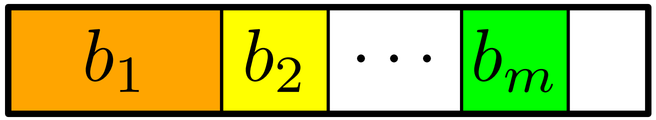
- $m = 2$: $b_1 + b_2 \geq 2/3$, so $b_1 \geq 1/3$.
  - If $b_2 \geq 1/3$, $W(b_1) + W(b_2) \geq 1$.
  - Else $W(b_1) + W(b_2) = \frac{6}{5}(b_1 + b_2) + \frac{3}{5} b_2$

$\Rightarrow$ $W(b_1) + W(b_2) \geq \frac{6}{5}(1 - \alpha) + \frac{3}{5}\alpha = \frac{6}{5} - \frac{3}{5}\alpha \geq 1$.

First Fit — First Step

Proof step 1: $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 1: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$

B_i 

If $\sum_{\ell=1}^{m} b_\ell > 1 - \alpha$, then $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$.

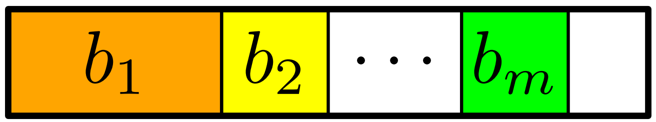
Proof for $\alpha \in (1/6, 1/3), m \geq 3$:

- $b_1 \geq 1/3$:
$$\sum_{\ell=1}^{m} W(b_\ell) \geq \frac{3b_2}{5} + \frac{6}{5}\sum_{\ell=1}^{m} b_\ell \geq \frac{6}{5}(1-\alpha) + \frac{3}{5}\alpha \geq 1.$$
- $b_1 < 1/3$:
$$\sum_{\ell=1}^{m} W(b_\ell) \geq \frac{9}{5}(b_1 + b_2) + \frac{6}{5}\sum_{\ell=3}^{m} b_\ell \geq \frac{3}{5}(2\alpha) + \frac{6}{5}(1-\alpha) - \frac{1}{5} \geq 1.$$

First Fit — First Step

Proof step 1: $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 1: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$

B_i


If $\sum_{\ell=1}^{m} b_\ell > 1 - \alpha$, then $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$.

Proof for $\alpha \in (1/6, 1/3), m \geq 3$:

- $b_1 \geq 1/3$:
$$\sum_{\ell=1}^{m} W(b_\ell) \geq \frac{3b_2}{5} + \frac{6}{5} \sum_{\ell=1}^{m} b_\ell \geq \frac{6}{5}(1-\alpha) + \frac{3}{5}\alpha \geq 1.$$
- $b_1 < 1/3$:
$$\sum_{\ell=1}^{m} W(b_\ell) \geq \frac{9}{5}(b_1 + b_2) + \frac{6}{5} \sum_{\ell=3}^{m} b_\ell \geq \frac{3}{5}(2\alpha) + \frac{6}{5}(1-\alpha) - \frac{1}{5} \geq 1.$$

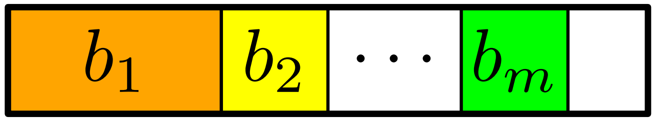
Claim 1: Done!

First Fit — First Step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 2: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$,
$\sum_{\ell=1}^{m} W(b_\ell) = 1 - \beta, \beta > 0.$

B_i



Then, either:

a) $m = 1, b_1 \leq 1/2$,

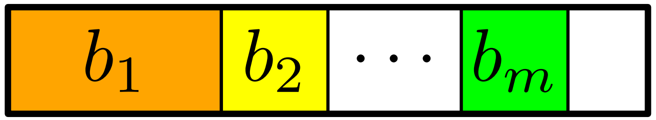
b) $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \frac{5\beta}{9}$.

First Fit — First Step

Proof step 1: $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 2: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$,

$$\sum_{\ell=1}^{m} W(b_\ell) = 1 - \beta, \beta > 0.$$

B_i 

Then, either:

a) $m = 1, b_1 \leq 1/2$,

b) $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \frac{5\beta}{9}$.

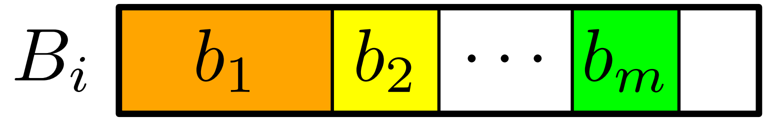
Proof: If a) does not hold: $m \geq 2$, $b_1 \geq b_2 \geq \alpha$.

By Claim 1: $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \gamma, \gamma \geq 0$. Build pseudo-bin:

First Fit — First Step

Proof step 1: $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 2: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$,
$\sum_{\ell=1}^{m} W(b_\ell) = 1 - \beta, \beta > 0.$

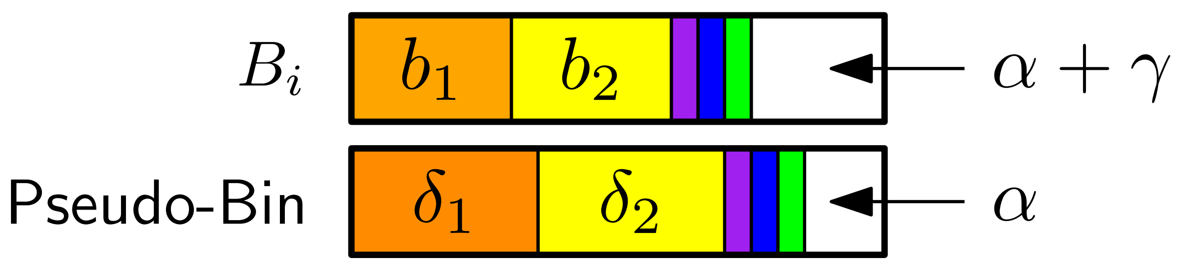


Then, either:

a) $m = 1, b_1 \leq 1/2$,

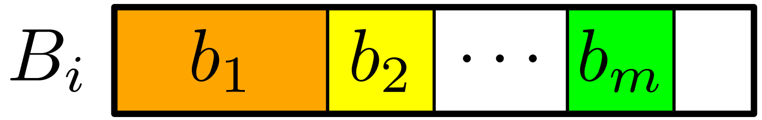
b) $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \frac{5\beta}{9}$.

Proof: If a) does not hold: $m \geq 2$, $b_1 \geq b_2 \geq \alpha$.
By Claim 1: $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \gamma, \gamma \geq 0$. Build pseudo-bin:



Proof step 1: $\mathrm{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 2: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$,
$\sum_{\ell=1}^{m} W(b_\ell) = 1 - \beta, \beta > 0.$

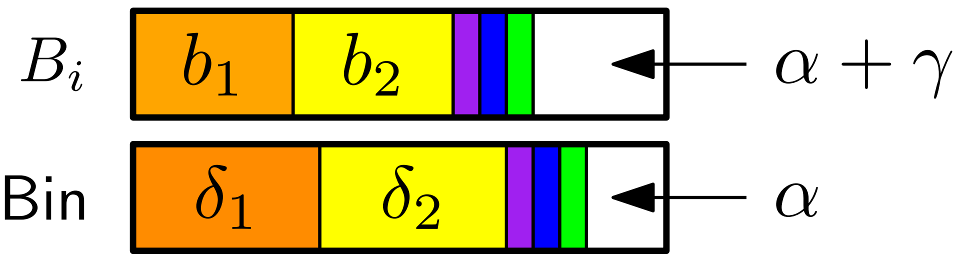


Then, either:

a) $m = 1, b_1 \leq 1/2,$

b) $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \frac{5\beta}{9}.$

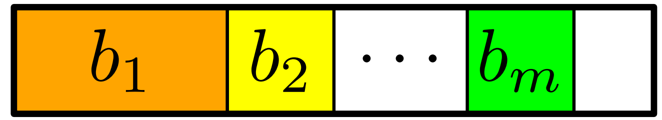
Proof: If a) does not hold: $m \geq 2$, $b_1 \geq b_2 \geq \alpha$.
By Claim 1: $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \gamma, \gamma \geq 0$. Build pseudo-bin:



$\delta_1 \geq b_1$
$\delta_2 \geq b_2$

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Claim 2: Bin B_i with coarseness $\alpha < 1/2$, $b_1 \geq b_2 \geq \cdots \geq b_m$,
$\sum_{\ell=1}^{m} W(b_\ell) = 1 - \beta, \beta > 0.$

B_i 

Then, either:

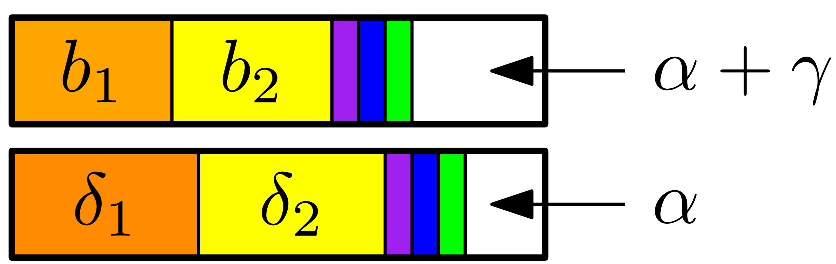
a) $m = 1, b_1 \leq 1/2$,

b) $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \frac{5\beta}{9}$.

Proof: If a) does not hold: $m \geq 2$, $b_1 \geq b_2 \geq \alpha$.
By Claim 1: $\sum_{\ell=1}^{m} b_\ell \leq 1 - \alpha - \gamma, \gamma \geq 0$. Build pseudo-bin:

B_i

Pseudo-Bin



$\delta_1 \geq b_1$

$\delta_2 \geq b_2$

$\delta_1, \delta_2 \leq 1/2$

Claim 1: $W(\delta_1) + W(\delta_2) + \sum_{\ell=3}^{m} W(b_\ell) \geq 1 \Rightarrow \gamma \geq \frac{5\beta}{9}$.
slope of W

First Fit — Finalizing First Step

Proof step 1: $\mathsf{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

First Fit — Finalizing First Step

Proof step 1: $\mathsf{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of First Fit.

First Fit — Finalizing First Step

Proof step 1: $\mathsf{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of $\textsc{First Fit}$.

For bins with weight $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$: Done.

First Fit — Finalizing First Step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of First Fit.

For bins with weight $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$: Done.

Consider bins $C_i, 1 \leq i \leq k$ with weight $\sum_{\ell=1}^{m} W(c_\ell) = 1 - \beta_i$.

First Fit — Finalizing First Step

Proof step 1: $\mathsf{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of $\textsc{First Fit}$.

For bins with weight $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$: Done.

Consider bins $C_i, 1 \leq i \leq k$ with weight $\sum_{\ell=1}^{m} W(c_\ell) = 1 - \beta_i$.

Coarseness of C_i: $\alpha_i < {}^1\!/_2$ (due to weight)

First Fit — Finalizing First Step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of First Fit.

For bins with weight $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$: Done.

Consider bins $C_i, 1 \leq i \leq k$ with weight $\sum_{\ell=1}^{m} W(c_\ell) = 1 - \beta_i$.

Coarseness of C_i: $\alpha_i < 1/2$ (due to weight)

Claim 2: $\sum_{\ell=1}^{m} c_\ell \leq 1 - \alpha_i - \frac{5}{9}\beta_i \Rightarrow \alpha_{i+1} \geq \alpha_i + \frac{5}{9}\beta_i$

First Fit — Finalizing First Step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of First Fit.

For bins with weight $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$: Done.

Consider bins $C_i, 1 \leq i \leq k$ with weight $\sum_{\ell=1}^{m} W(c_\ell) = 1 - \beta_i$.

Coarseness of C_i: $\alpha_i < 1/2$ (due to weight)

Claim 2: $\sum_{\ell=1}^{m} c_\ell \leq 1 - \alpha_i - \frac{5}{9}\beta_i \Rightarrow \alpha_{i+1} \geq \alpha_i + \frac{5}{9}\beta_i$

Therefore β_i is bounded: $\beta_i \leq \frac{9}{5}(\alpha_{i+1} - \alpha_i)$

First Fit — Finalizing First Step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of First Fit.

For bins with weight $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$: Done.

Consider bins $C_i, 1 \leq i \leq k$ with weight $\sum_{\ell=1}^{m} W(c_\ell) = 1 - \beta_i$.

Coarseness of C_i: $\alpha_i < 1/2$ (due to weight)

Claim 2: $\sum_{\ell=1}^{m} c_\ell \leq 1 - \alpha_i - \frac{5}{9}\beta_i \Rightarrow \alpha_{i+1} \geq \alpha_i + \frac{5}{9}\beta_i$

Therefore β_i is bounded: $\beta_i \leq \frac{9}{5}(\alpha_{i+1} - \alpha_i)$

$\sum_{i=1}^{k-1} \beta_i \leq \frac{9}{5} \sum_{i=2}^{k} (\alpha_i - \alpha_{i-1}) = \frac{9}{5}(\alpha_k - \alpha_1) \leq \frac{9}{5} \cdot \frac{1}{2} < 1.$

First Fit — Finalizing First Step

Proof step 1: $\text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 2$

Let $B_1, \ldots, B_m$ be the bins of First Fit.

For bins with weight $\sum_{\ell=1}^{m} W(b_\ell) \geq 1$: Done.

Consider bins $C_i, 1 \leq i \leq k$ with weight $\sum_{\ell=1}^{m} W(c_\ell) = 1 - \beta_i$.

Coarseness of C_i: $\alpha_i < 1/2$ (due to weight)

Claim 2: $\sum_{\ell=1}^{m} c_\ell \leq 1 - \alpha_i - \frac{5}{9}\beta_i \Rightarrow \alpha_{i+1} \geq \alpha_i + \frac{5}{9}\beta_i$

Therefore β_i is bounded: $\beta_i \leq \frac{9}{5}(\alpha_{i+1} - \alpha_i)$

$\sum_{i=1}^{k-1} \beta_i \leq \frac{9}{5} \sum_{i=2}^{k} (\alpha_i - \alpha_{i-1}) = \frac{9}{5}(\alpha_k - \alpha_1) \leq \frac{9}{5} \cdot \frac{1}{2} < 1.$

$\beta_k \leq 1 \Rightarrow \text{FF}(\sigma) \leq \sum_{i=1}^{n} W(a_i) + 1 + 1.$

First Fit — Second Step

Second step: $\sum_{i=1}^{n} W(a_i) \leq 1.7 \cdot \mathsf{OPT}$

First Fit — Second Step

Second step: $\sum_{i=1}^{n} W(a_i) \leq 1.7 \cdot \mathrm{OPT}$

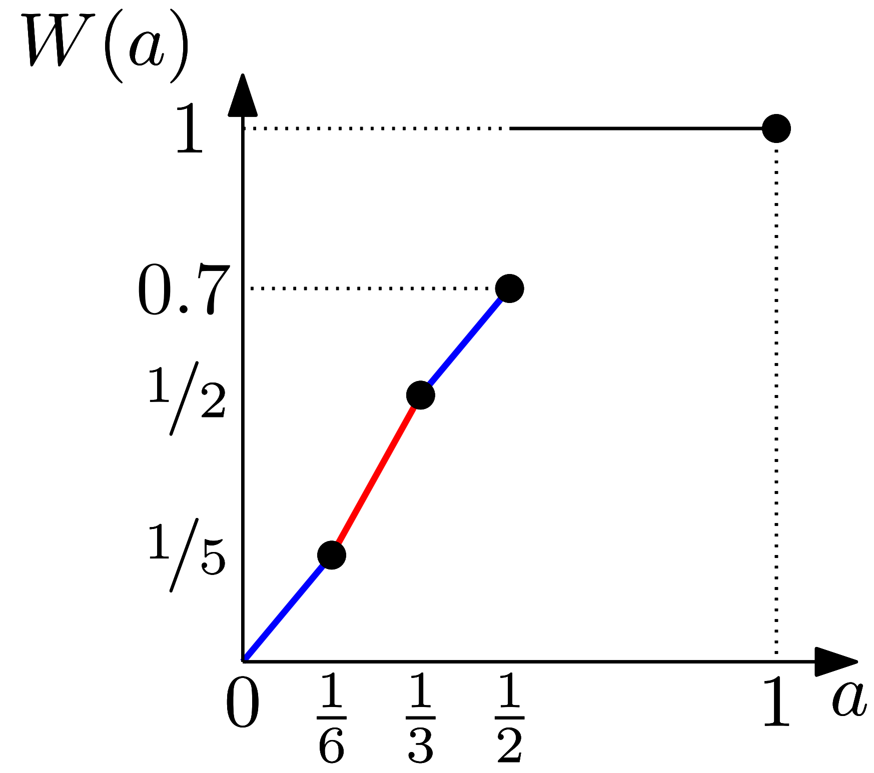
Consider OPT's bins

First Fit — Second Step

Second step: $\sum_{i=1}^{n} W(a_i) \leq 1.7 \cdot \text{OPT}$

Consider OPT's bins

For any possible bin $b_1, \ldots, b_k$:
$\sum_{i=1}^{k} W(b_i) \leq 1.7$



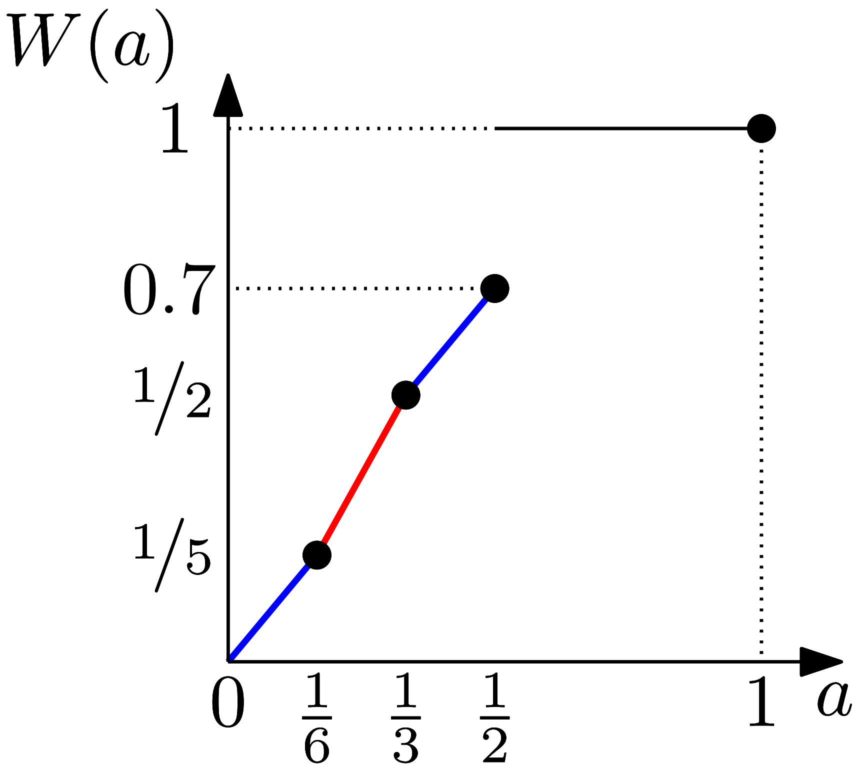
First Fit — Second Step

Second step: $\sum_{i=1}^{n} W(a_i) \leq 1.7 \cdot \text{OPT}$

Consider OPT's bins

For any possible bin $b_1, \ldots, b_k$:
$\sum_{i=1}^{k} W(b_i) \leq 1.7$

All $b_i \leq 1/2 : W(b_i) \leq \frac{3b_i}{2}$.



First Fit — Second Step

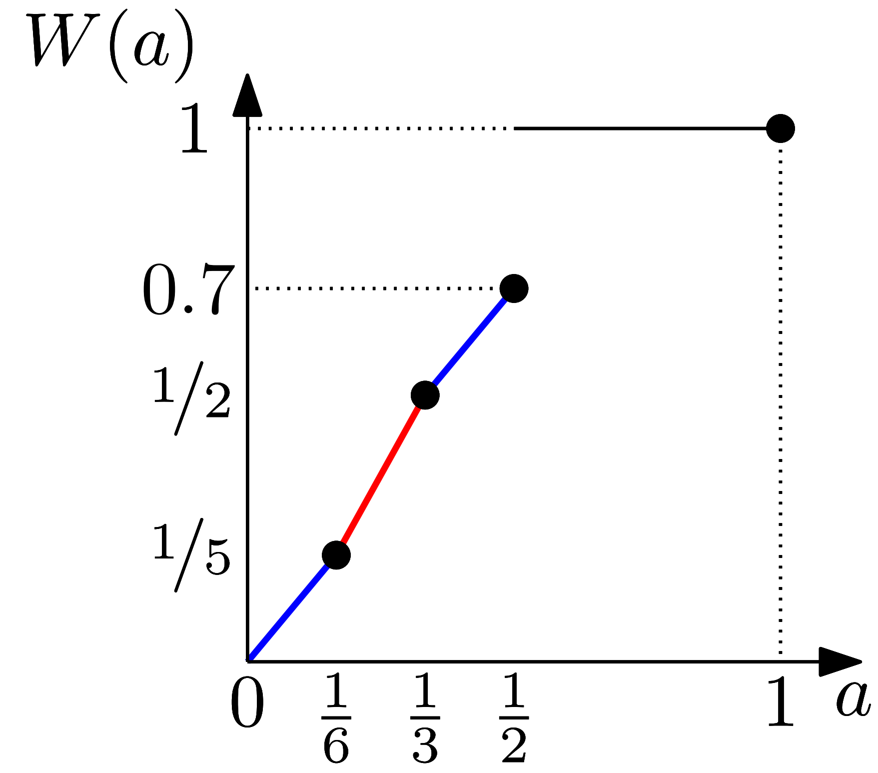
Second step: $\sum_{i=1}^{n} W(a_i) \leq 1.7 \cdot \text{OPT}$

Consider OPT's bins

For any possible bin $b_1, \ldots, b_k$:
$\sum_{i=1}^{k} W(b_i) \leq 1.7$

All $b_i \leq 1/2 : W(b_i) \leq \frac{3b_i}{2}$.

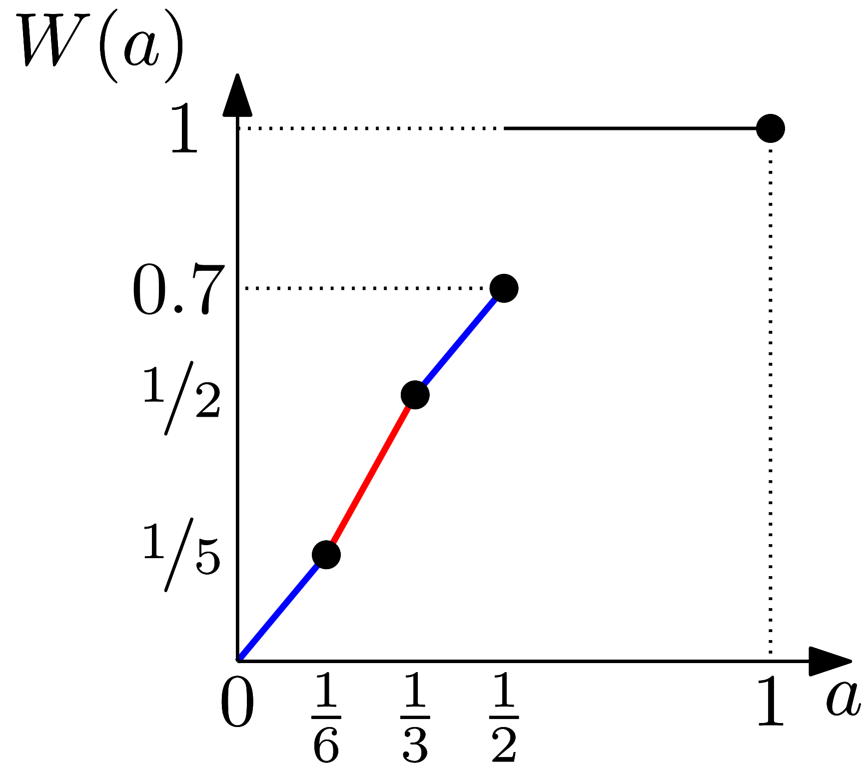
Thus, assume at least $b_1 > 1/2$



First Fit — Second Step

Second step: $\sum_{i=1}^{n} W(a_i) \leq 1.7 \cdot \mathrm{OPT}$

Consider OPT's bins



For any possible bin $b_1, \ldots, b_k$:
$\sum_{i=1}^{k} W(b_i) \leq 1.7$

All $b_i \leq 1/2 : W(b_i) \leq \frac{3b_i}{2}$.

Thus, assume at least $b_1 > 1/2$

Reduce possible configurations:

- Item $b_i \in (1/3, 1/2) \to \delta_i = 1/3, \delta_i - 1/3 < 1/6$
- Items $b_i, b_\ell \leq 1/6 \to \varepsilon_i = b_i + b_\ell$
- At most one $b_i \in (0, 1/6]$, all others in $(1/6, 1/3]$
- At most 4 items; simple case distinction

First Fit — Recap

We have seen:

- $\mathsf{FF}(\sigma) \leq 1.7 \cdot \mathsf{OPT} + 2$
- First Fit is (asymptotically) 1.7-competitive
- No Any Fit-algorithm can be better

First Fit — Recap

We have seen:

- $\mathsf{FF}(\sigma) \leq 1.7 \cdot \mathsf{OPT} + 2$
- First Fit is (asymptotically) 1.7-competitive
- No Any Fit-algorithm can be better

What's still missing?

- Better algorithm (must open bins before full)
- More general lower bound

How can we trick an(y) online algorithm?

Idea to trick algorithm:

- Send in group of small items
- If the algorithm packs them together:
  - Send in bigger items
- Otherwise:
  - End the sequence
- Repeat in phases, items getting bigger

Bin Packing — Lower Bounds

How can we trick an(y) online algorithm?

Idea to trick algorithm:

- Send in group of small items
- If the algorithm packs them together:
  - Send in bigger items
- Otherwise:
  - End the sequence
- Repeat in phases, items getting bigger

Bin Packing — Lower Bounds

Let's try this idea for some competitive ratio c:

- Pick k (scaling factor)
- Phase I: Send in $6k \times ({}^1\!/_7 + \varepsilon)$
- If algorithm uses more than ck bins: we stop.

Bin Packing — Lower Bounds

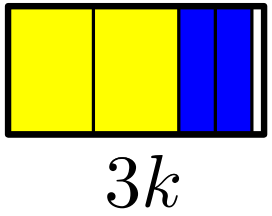
Let's try this idea for some competitive ratio c:

- Pick k (scaling factor)
- Phase I: Send in $6k \times (1/7 + \varepsilon)$
- If algorithm uses more than ck bins: we stop.
- Phase II: Send in $6k \times (1/3 + \varepsilon)$
- What would OPT do?

Bin Packing — Lower Bounds

Let's try this idea for some competitive ratio c:

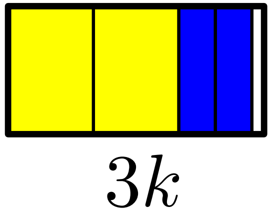
- Pick k (scaling factor)
- Phase I: Send in $6k \times (1/7 + \varepsilon)$
- If algorithm uses more than ck bins: we stop.
- Phase II: Send in $6k \times (1/3 + \varepsilon)$
- What would OPT do?



Bin Packing — Lower Bounds

Let's try this idea for some competitive ratio c:

- Pick k (scaling factor)
- Phase I: Send in $6k \times (1/7 + \varepsilon)$
- If algorithm uses more than ck bins: we stop.
- Phase II: Send in $6k \times (1/3 + \varepsilon)$
- What would OPT do?

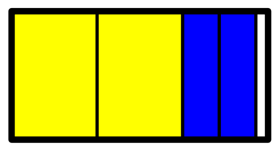


- If the algorithm uses more than $3ck$ bins: we stop.

Bin Packing — Lower Bounds

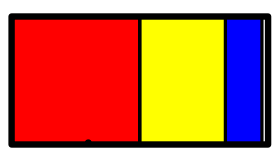
Let's try this idea for some competitive ratio c:

- Pick k (scaling factor)
- Phase I: Send in $6k \times (1/7 + \varepsilon)$
- If algorithm uses more than ck bins: we stop.
- Phase II: Send in $6k \times (1/3 + \varepsilon)$
- What would OPT do?



$3k$

- If the algorithm uses more than $3ck$ bins: we stop.
- Phase III: Send in $6k \times (1/2 + \varepsilon)$
- Algorithm must use at most $6ck$ bins

OPT 

$6k$

Sensible bin configurations of the algorithm:

Sensible bin configurations of the algorithm:

$C_1 : (6, 0, 0)$ — 6 I items

Sensible bin configurations of the algorithm:

$C_1 : (6, 0, 0)$ — 6 I items $C_2 : (4, 1, 0)$ — 4 I, 1 II items

$C_3 : (3, 1, 0)$ $C_4 : (2, 2, 0)$ $C_5 : (5, 0, 0)$

$C_6 : (1, 1, 1)$ $C_7 : (3, 0, 1)$ $C_8 : (2, 0, 1)$

$C_9 : (0, 1, 1)$ $C_{10} : (1, 2, 0)$ $C_{11} : (0, 2, 0)$

$C_{12} : (0, 0, 1)$

Bin Packing — Lower Bounds

Sensible bin configurations of the algorithm:

$C_1 : (6, 0, 0)$ — 6 I items $\quad$ $C_2 : (4, 1, 0)$ — 4 I, 1 II items

$C_3 : (3, 1, 0)$ $\quad$ $C_4 : (2, 2, 0)$ $\quad$ $C_5 : (5, 0, 0)$

$C_6 : (1, 1, 1)$ $\quad$ $C_7 : (3, 0, 1)$ $\quad$ $C_8 : (2, 0, 1)$

$C_9 : (0, 1, 1)$ $\quad$ $C_{10} : (1, 2, 0)$ $\quad$ $C_{11} : (0, 2, 0)$

$C_{12} : (0, 0, 1)$

Variable k_i: Number of bins with C_i after phase III, c CR

Bin Packing — Lower Bounds

Sensible bin configurations of the algorithm:

$C_1 : (6,0,0)$ — 6 I items $\quad$ $C_2 : (4,1,0)$ — 4 I, 1 II items

$C_3 : (3,1,0)$ $\quad$ $C_4 : (2,2,0)$ $\quad$ $C_5 : (5,0,0)$

$C_6 : (1,1,1)$ $\quad$ $C_7 : (3,0,1)$ $\quad$ $C_8 : (2,0,1)$

$C_9 : (0,1,1)$ $\quad$ $C_{10} : (1,2,0)$ $\quad$ $C_{11} : (0,2,0)$

$C_{12} : (0,0,1)$

Variable k_i: Number of bins with C_i after phase III, c CR

Phase I:

$$6k_1 + 4k_2 + 3k_3 + 2k_4 + 5k_5 + k_6 + 3k_7 + 2k_8 + k_{10} = 6k$$

$$k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_{10} \leq ck$$

Phase II:

$$k_2 + k_3 + 2k_4 + k_6 + k_9 + 2k_{10} + 2k_{11} = 6k$$

$$\sum_{i=1}^{11} k_i \leq 3ck$$

Phase III:

$$k_6 + k_7 + k_8 + k_9 + k_{12} = 6k$$

$$\sum_{i=1}^{12} k_i \leq 6ck$$

Bin Packing — Lower Bound

Goal: $\min c$ subject to

$6k_1 + 4k_2 + 3k_3 + 2k_4 + 5k_5 + k_6 + 3k_7 + 2k_8 + k_{10} = 6k$

$k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_{10} \leq ck$

$k_2 + k_3 + 2k_4 + k_6 + k_9 + 2k_{10} + 2k_{11} = 6k$

$\sum_{i=1}^{11} k_i \leq 3ck$

$k_6 + k_7 + k_8 + k_9 + k_{12} = 6k$

$\sum_{i=1}^{12} k_i \leq 6ck$

Bin Packing — Lower Bound

Goal: $\min c$ subject to

$6k_1 + 4k_2 + 3k_3 + 2k_4 + 5k_5 + k_6 + 3k_7 + 2k_8 + k_{10} = 6k$
$k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_{10} \leq ck$
$k_2 + k_3 + 2k_4 + k_6 + k_9 + 2k_{10} + 2k_{11} = 6k$
$\sum_{i=1}^{11} k_i \leq 3ck$
$k_6 + k_7 + k_8 + k_9 + k_{12} = 6k$
$\sum_{i=1}^{12} k_i \leq 6ck$

This is a linear program (LP)!

Bin Packing — Lower Bound

Goal: $\min c$ subject to

$$6k_1 + 4k_2 + 3k_3 + 2k_4 + 5k_5 + k_6 + 3k_7 + 2k_8 + k_{10} = 6k$$
$$k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_{10} \leq ck$$
$$k_2 + k_3 + 2k_4 + k_6 + k_9 + 2k_{10} + 2k_{11} = 6k$$
$$\sum_{i=1}^{11} k_i \leq 3ck$$
$$k_6 + k_7 + k_8 + k_9 + k_{12} = 6k$$
$$\sum_{i=1}^{12} k_i \leq 6ck$$

This is a linear program (LP)!

We can divide by k (then the k_i are relative to k)

Bin Packing — Lower Bound

Goal: $\min c$ subject to

$6k_1 + 4k_2 + 3k_3 + 2k_4 + 5k_5 + k_6 + 3k_7 + 2k_8 + k_{10} = 6k$

$k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_{10} \leq ck$

$k_2 + k_3 + 2k_4 + k_6 + k_9 + 2k_{10} + 2k_{11} = 6k$

$\sum_{i=1}^{11} k_i \leq 3ck$

$k_6 + k_7 + k_8 + k_9 + k_{12} = 6k$

$\sum_{i=1}^{12} k_i \leq 6ck$

This is a linear program (LP)!

We can divide by k (then the k_i are relative to k)

This gives us a lower bound on c for any algorithm!

Bin Packing — Lower Bound

Goal: $\min c$ subject to

$6k_1 + 4k_2 + 3k_3 + 2k_4 + 5k_5 + k_6 + 3k_7 + 2k_8 + k_{10} = 6k$
$k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_{10} \leq ck$
$k_2 + k_3 + 2k_4 + k_6 + k_9 + 2k_{10} + 2k_{11} = 6k$
$\sum_{i=1}^{11} k_i \leq 3ck$
$k_6 + k_7 + k_8 + k_9 + k_{12} = 6k$
$\sum_{i=1}^{12} k_i \leq 6ck$

This is a linear program (LP)!

We can divide by k (then the k_i are relative to k)

This gives us a lower bound on c for any algorithm!

3 phases: $c \geq 1.5$; more phases improve that bound

Bin Packing — Lower Bound

Goal: $\min c$ subject to

$6k_1 + 4k_2 + 3k_3 + 2k_4 + 5k_5 + k_6 + 3k_7 + 2k_8 + k_{10} = 6k$
$k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_{10} \leq ck$
$k_2 + k_3 + 2k_4 + k_6 + k_9 + 2k_{10} + 2k_{11} = 6k$
$\sum_{i=1}^{11} k_i \leq 3ck$
$k_6 + k_7 + k_8 + k_9 + k_{12} = 6k$
$\sum_{i=1}^{12} k_i \leq 6ck$

This is a linear program (LP)!

We can divide by k (then the k_i are relative to k)

This gives us a lower bound on c for any algorithm!

3 phases: $c \geq 1.5$; more phases improve that bound

Item sizes for more phases: $\frac{1}{2} + \frac{1}{3} + \frac{1}{7} = \frac{41}{42}$; phase IV: $\frac{1}{43}$
$\frac{1}{2} + \frac{1}{3} + \frac{1}{7} + \frac{1}{43} = \frac{1805}{1806}$; phase V: $\frac{1}{1807}$

Bin Packing — Lower Bound

Van Vliet (1992):
Manual (closed-form) solution: $c \to 1.5401\ldots$ for #phases $\to \infty$

Theorem 6. **(Van Vliet [106, 107])** *For any on-line bin packing algorithm A, the bound $R_A^\infty \geq 1.5401$ holds.*

106. A. Van Vliet. An improved lower bound for on-line bin packing algorithms. *Inform. Process. Lett.*, 43:277–284, 1992.
107. A. Van Vliet. *Lower and upper bounds for on-line bin packing and scheduling heuristics*. PhD thesis, Erasmus University, Rotterdam, The Netherlands, 1995.