Online Algorithms
Lecture 3

Jiří Sgall

Computer Science Institute of the Charles Univ., Praha

EWSCS, Palmse, March 2020

A Brief Review of Bin Packing

Bin packing

- Input: Sequence of **items** $a_1, \ldots, a_n \in [0, 1]$.
- Output: Assign into **bins** of size 1.
- Objective: Minimize the number of bins.

Complexity results

- It is NP-hard to decide if $\mathrm{OPT}(I) = 2$.
  Thus it is NP-hard to approximate with ratio $< 3/2$.
- There exists a $3/2$-approximation algorithm.
- There exists an asymptotic approximation scheme.
  I.e., in polynomial time we can pack the items into $(1 + \varepsilon)\mathrm{OPT}(I) + 1$ bins.
- In poly time we can pack into $\mathrm{OPT}(I) + \log(\mathrm{OPT}(I))$ bins.

Performance measures

Absolute approximation ratio

For each instance I, the algorithm gives

$$ALG(I) \leq R \cdot \mathrm{OPT}(I)$$

Asymptotic approximation ratio

There exists a constant C such that for each instance I, the algorithm gives

$$ALG(I) \leq R \cdot \mathrm{OPT}(I) + C$$

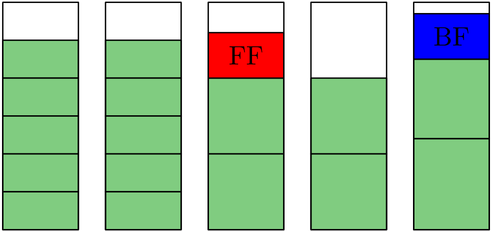
Two online algorithms

First Fit

Packs items one by one, always into the **first** bin where it fits.
Opens a new bin only when necessary.

Best Fit

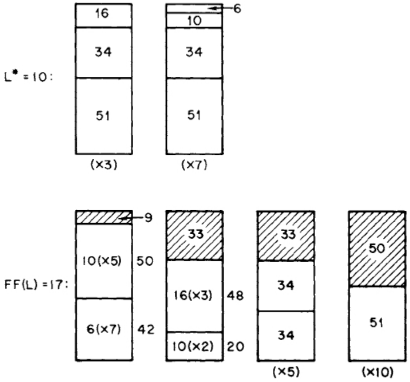
Packs items one by one, always into the **most full** bin where it fits.
Opens a new bin only when necessary.



Known Results

Classics [Ullman, 1971][Garey et al, 1973]

The asymptotic approximation ratio of both First Fit and Best Fit is equal to 1.7. More precisely, $\mathrm{FF}, \mathrm{BF} \le \lceil 1.7 \cdot \mathrm{OPT} \rceil$.
Example with $\mathrm{FF} = \mathrm{BF} = 17$ and $\mathrm{OPT} = 10$.



Bin size is 101.

Known Results

Classics [Ullman, 1971][Garey et al, 1973]

The asymptotic approximation ratio of both First Fit and Best Fit is equal to 1.7. More precisely, $\mathrm{FF}, \mathrm{BF} \leq \lceil 1.7 \cdot \mathrm{OPT} \rceil$.
Example with $\mathrm{FF} = \mathrm{BF} = 17$ and $\mathrm{OPT} = 10$.
Example with $\mathrm{FF} = \mathrm{BF} = 17 \cdot k$ and $\mathrm{OPT} = 10 \cdot k + 1$.

Known Results

Classics [Ullman, 1971][Garey et al, 1973]

The asymptotic approximation ratio of both First Fit and Best Fit is equal to 1.7. More precisely, $\mathrm{FF}, \mathrm{BF} \leq \lceil 1.7 \cdot \mathrm{OPT} \rceil$.
Example with $\mathrm{FF} = \mathrm{BF} = 17$ and $\mathrm{OPT} = 10$.
Example with $\mathrm{FF} = \mathrm{BF} = 17 \cdot k$ and $\mathrm{OPT} = 10 \cdot k + 1$.

Absolute ratio [Simchi-Levi, 1994]

$\mathrm{FF}, \mathrm{BF} \leq 1.75 \cdot \mathrm{OPT}$.

Recent improvements [Xia,Tan, 2010][Boyar et al, 2012][Neméth, 2011]

$\mathrm{FF} \leq 1.7 \cdot \mathrm{OPT} + 0.7$ and $\mathrm{FF} \leq 1.7119 \cdot \mathrm{OPT}$.
Uses analysis of cases with small OPT.

First Fit Decreasing [Johnson, 1973][Dósa, 2007]

$\mathrm{FFD} \leq \frac{11}{9} \cdot \mathrm{OPT} + \frac{2}{3}$, and this is optimal for all values of OPT.

Main Technique

Classical technique: **Weight functions**.

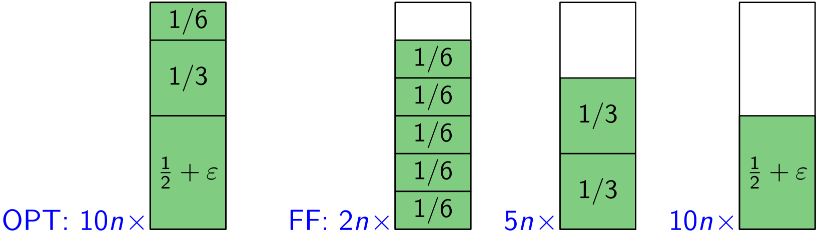
Find a weight of items such that

- Each bin in OPT has weight ≤ 1.7.
- Each bin in FF (BF) has weight ≥ 1 on average.

Combine weight functions with **amortized analysis**.

Idealized example for First Fit

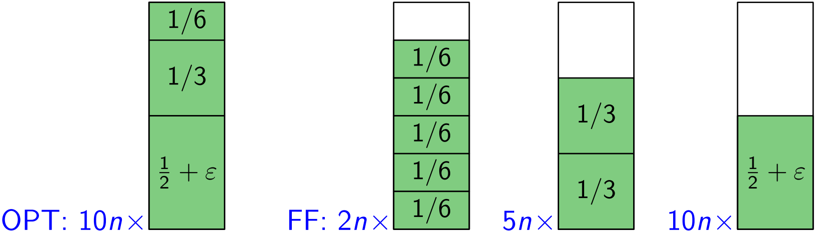
Assume that the algorithm cannot have bins of size exactly 1.



Essentially, this can be achieved by changing the item sizes by a small amount.

Idealized example for First Fit

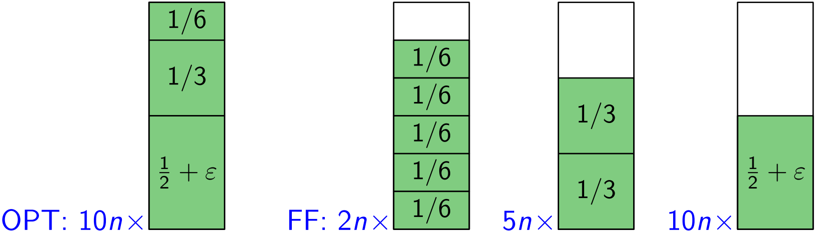
Assume that the algorithm cannot have bins of size exactly 1.



What should be the weights?

Idealized example for First Fit

Assume that the algorithm cannot have bins of size exactly 1.



$w(1/6) = 0.2 \quad w(1/3) = 0.5 \quad w(\frac{1}{2}+\varepsilon) = 1$

The weight function

- Weight: **Scaled size** plus a **bonus.**
- $w(a) = \frac{6}{5}a + b(a)$

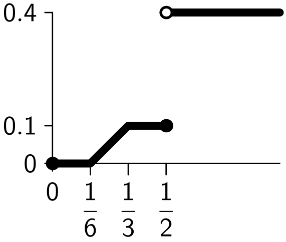
The weight function

- Weight: **Scaled size** plus a **bonus.**

- $w(a) = \frac{6}{5}a + b(a)$

- $b(a) =$

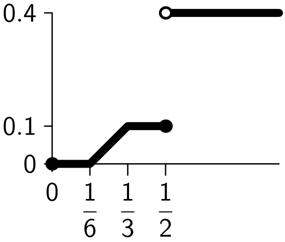
$$\begin{cases} 0 & \text{if } a \le \frac{1}{6} \\ \frac{3}{5}\left(a - \frac{1}{6}\right) & \text{if } a \in [\frac{1}{6}, \frac{1}{3}] \\ 0.1 & \text{if } a \in [\frac{1}{3}, \frac{1}{2}) \\ 0.4 & \text{if } a > \frac{1}{2} \end{cases}$$



Offline bins

Each bin (a set of items of size ≤ 1) contains **bonus items**:

- either no item of size $> 1/2$ and at most 5 items with bonus at most 0.1 each (actually the total is ≤ 0.3),
- or one item of size $> 1/2$ and at most 2 items with bonus at most 0.1 total.
- Thus the **total bonus** is at most 0.5;



Offline bins

Each bin (a set of items of size ≤ 1) contains **bonus items**:

- either no item of size $> 1/2$ and at most 5 items with bonus at most 0.1 each (actually the total is ≤ 0.3),
- or one item of size $> 1/2$ and at most 2 items with bonus at most 0.1 total.
- Thus the **total bonus** is at most 0.5;
- the **total scaled size** is at most 1.2;
- the **total weight** is at most 1.7.

First Fit bins

- No item in a later bin fits into any previous bin.
- There is at most one bin of size $\leq 1/2$.
- There is at most one bin of size $\leq 2/3$ with at least two items.

First Fit bins

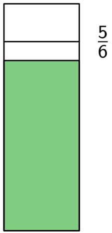
- No item in a later bin fits into any previous bin.
- There is at most one bin of size $\leq 1/2$.
- There is at most one bin of size $\leq 2/3$ with at least two items.
- Bins of size $\geq 5/6$ have weight ≥ 1.
- Bins with an item of size $> 1/2$ have weight ≥ 1.

First Fit bins

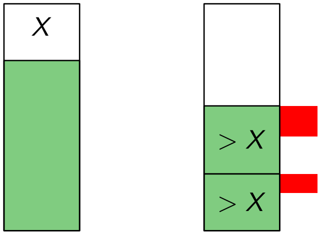
- No item in a later bin fits into any previous bin.
- There is at most one bin of size $\leq 1/2$.
- There is at most one bin of size $\leq 2/3$ with at least two items.
- Bins of size $\geq 5/6$ have weight ≥ 1.
- Bins with an item of size $> 1/2$ have weight ≥ 1.
- For the remaining bins with sizes in $(2/3, 5/6)$ we use **amortization**.
  We show that the scaled size of a bin plus the bonus of the **following** such bin is ≥ 1.

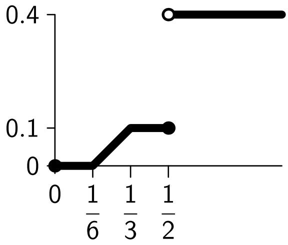
Amortization

For each bin with size in $(2/3, 5/6)$, at least two items, and no item $> 1/2$ we show that the scaled size of this bin plus the bonus of the **following** such bin is ≥ 1.

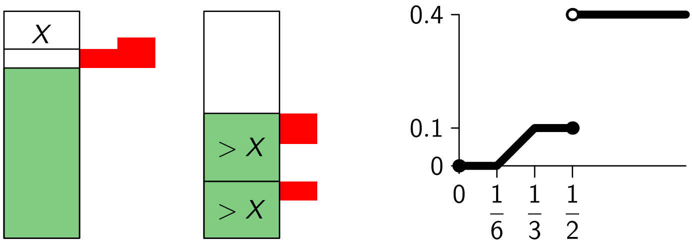


For each bin with size in $(2/3, 5/6)$, at least two items, and no item $> 1/2$ we show that the scaled size of this bin plus the bonus of the **following** such bin is ≥ 1.





For each bin with size in $(2/3, 5/6)$, at least two items, and no item $> 1/2$ we show that the scaled size of this bin plus the bonus of the **following** such bin is ≥ 1.



The result

$$\mathrm{FF}(I) - 3 < w(I) \leq 1.7 \cdot \mathrm{OPT}(I)$$

Theorem

First Fit has **asymptotic** approximation ratio 1.7.

$$\mathrm{FF}(I) < 1.7 \cdot \mathrm{OPT}(I) + 3$$

The result

$$FF(I) - 3 < w(I) \le 1.7 \cdot OPT(I)$$

Theorem

First Fit has **asymptotic** approximation ratio 1.7.

$$FF(I) < 1.7 \cdot OPT(I) + 3$$

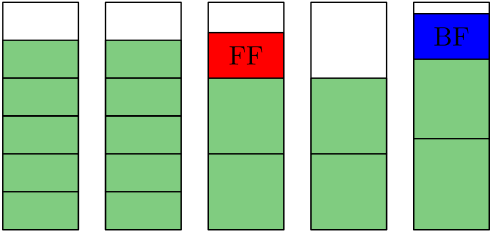
With some more work...

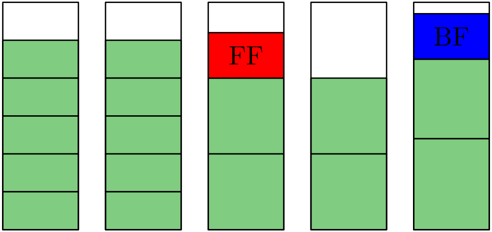
Theorem

First Fit and **Best Fit** have **absolute** approximation ratio 1.7.

$$FF(I), BF(I) \le 1.7 \cdot OPT(I)$$

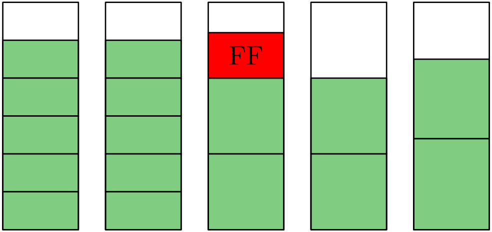
First Fit vs. Best Fit





First Fit is a special case of Best Fit.

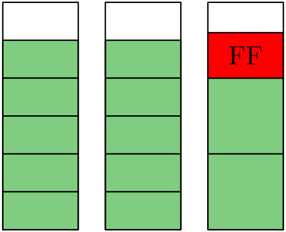
First Fit vs. Best Fit



First Fit is a special case of Best Fit.

- Suppose we have a bad example for First Fit. No item fits into any previous bin.

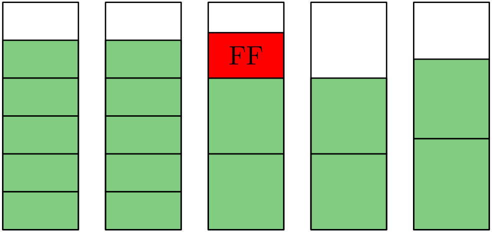
First Fit vs. Best Fit



First Fit is a special case of Best Fit.

- Suppose we have a bad example for First Fit. No item fits into any previous bin.
- Reorder the instance so that first the items from the first bin arrive, then the items from the next bin, etc.

First Fit vs. Best Fit

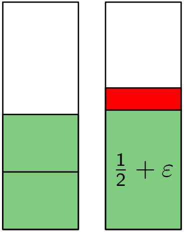


First Fit is a special case of Best Fit.

- Suppose we have a bad example for First Fit. No item fits into any previous bin.
- Reorder the instance so that first the items from the first bin arrive, then the items from the next bin, etc.
- The resulting First Fit and Best Fit packings are the same.

Best Fit revisited

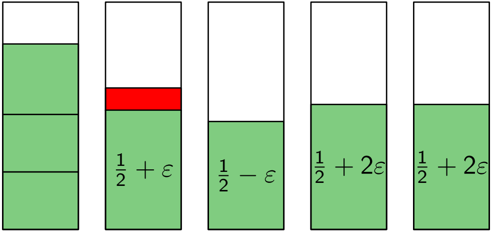
Best Fit can create packings very different from First Fit packings.



- We need some structural properties of Best Fit.

Best Fit revisited

Best Fit can create packings very different from First Fit packings.



- We need some structural properties of Best Fit.
- Some remaining cases are difficult.

Other good algorithms

2-Bounded Space Best Fit (BBF_2)

Packs items as Best Fit into open bins.
It is allowed to close the most full bin if at least two bins remain.

Other good algorithms

2-Bounded Space Best Fit (BBF_2)

Packs items as Best Fit into open bins.
It is allowed to close the most full bin if at least two bins remain.

Almost Any Fit (AAF)

Packs items into any bin where it fits, with the exception of the smallest bin if the item fits into some previous bin.
Opens a new bin only when necessary.

Other good algorithms

2-Bounded Space Best Fit (BBF_2)

Packs items as Best Fit into open bins.
It is allowed to close the most full bin if at least two bins remain.

Almost Any Fit (AAF)

Packs items into any bin where it fits, with the exception of the smallest bin if the item fits into some previous bin.
Opens a new bin only when necessary.

Generalized Almost Any Fit (GAAF)

Packs items into any bin where it fits, with the exception of the (single) bin with size at most $1/2$ if the item fits into some previous bin. Opens a new bin only when necessary.

Other good algorithms

2-Bounded Space Best Fit (BBF_2)

Packs items as Best Fit into open bins.
It is allowed to close the most full bin if at least two bins remain.

Almost Any Fit (AAF)

Packs items into any bin where it fits, with the exception of the smallest bin if the item fits into some previous bin.
Opens a new bin only when necessary.

Generalized Almost Any Fit (GAAF)

Packs items into any bin where it fits, with the exception of the (single) bin with size at most $1/2$ if the item fits into some previous bin. Opens a new bin only when necessary.

All of these are asymptotically 1.7-competitive.

Other algorithms?

Is it always possible to remove the additive constant?

Other algorithms?

Is it always possible to remove the additive constant?

Counterexample

There exists a GAAF algorithm which is an asymptotic 1.7-approximation but not an absolute 1.7-approximation.

Counterexample

BBF_2 is not an absolute 1.7-approximation.

Other algorithms?

Is it always possible to remove the additive constant?

Counterexample

There exists a GAAF algorithm which is an asymptotic 1.7-approximation but not an absolute 1.7-approximation.

Counterexample

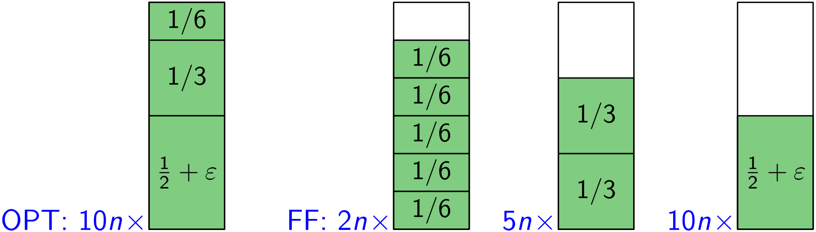
BBF_2 is not an absolute 1.7-approximation.

First Fit Decreasing [Johnson, 1973][Dósa, 2007]

$\mathrm{FFD} \leq \frac{11}{9} \cdot \mathrm{OPT} + \frac{2}{3}$, and this is optimal for all values of OPT.

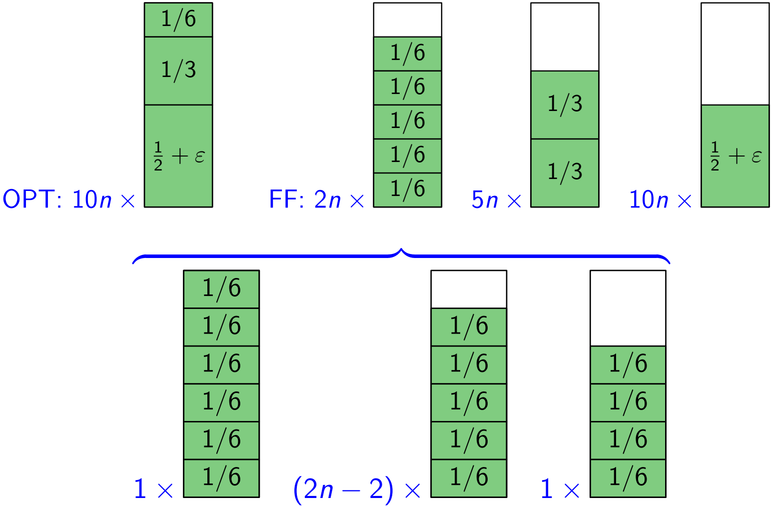
Idealized example revisited

Assume that the algorithm cannot have bins of size exactly 1.

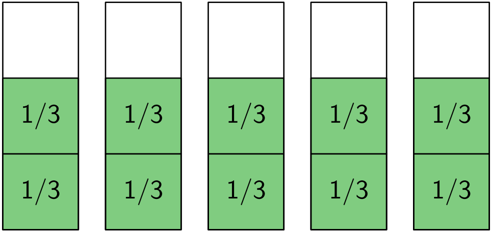


This can be achieved by changing the item sizes by a small amount and allowing OPT one extra bin.

Improved example

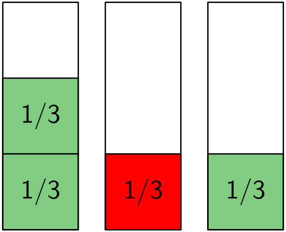


Optimal absolute ratio

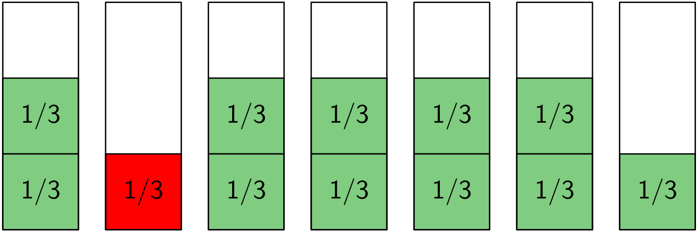


- We need to avoid bins with two items and size about $2/3$.

Optimal absolute ratio

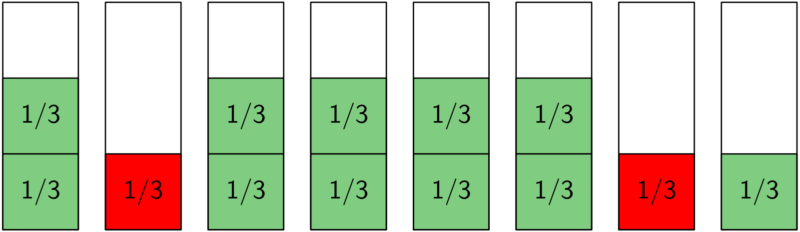


- We need to avoid bins with two items and size about $2/3$.
- Instead of 2nd such bin, we split the items and make a special bin.



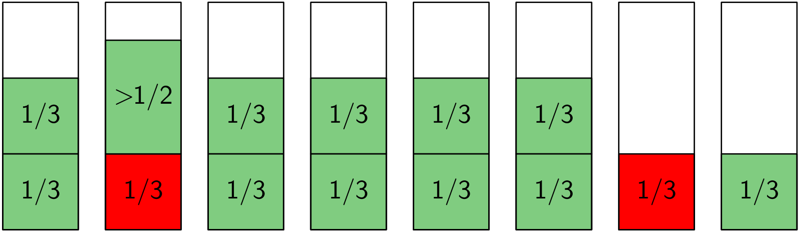
- We need to avoid bins with two items and size about $2/3$.
- Instead of 2nd such bin, we split the items and make a special bin.

Optimal absolute ratio



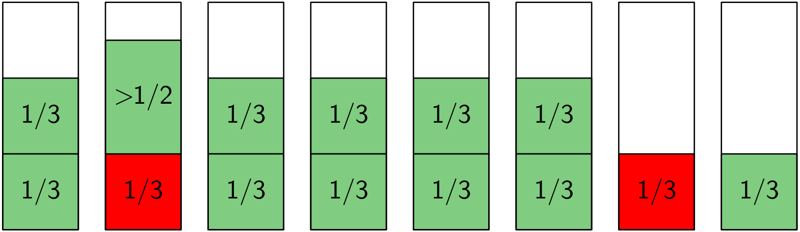
- We need to avoid bins with two items and size about 2/3.
- Instead of 2nd such bin, we split the items and make a special bin.
- Also instead of 6th, 10th, 14th, . . . , such bin
- "such bin" is actually a more technical notion.

Optimal absolute ratio



- We need to avoid bins with two items and size about $2/3$.
- Instead of 2nd such bin, we split the items and make a special bin.
- Also instead of 6th, 10th, 14th, ..., such bin
- "such bin" is actually a more technical notion.
- The special bins are reserved for large items.

Optimal absolute ratio



Theorem

The approximation ratio of the algorithm is $5/3$ and this is optimal among online algorithms.

Optimal absolute ratio – Analysis

Theorem

The approximation ratio of the algorithm is $5/3$ and this is optimal among online algorithms.

Three cases:

Optimal absolute ratio – Analysis

Theorem

The approximation ratio of the algorithm is $5/3$ and this is optimal among online algorithms.

Three cases:

- No special bins: Relatively easy, like FF with only a few $1/3$'s.

Optimal absolute ratio – Analysis

Theorem

The approximation ratio of the algorithm is $5/3$ and this is optimal among online algorithms.

Three cases:

- No special bins: Relatively easy, like FF with only a few $1/3$'s.
- All special bins have a large item: Weight functions similar to the analysis of FF.

Optimal absolute ratio – Analysis

Theorem

The approximation ratio of the algorithm is $5/3$ and this is optimal among online algorithms.

Three cases:

- No special bins: Relatively easy, like FF with only a few $1/3$'s.
- All special bins have a large item: Weight functions similar to the analysis of FF.
- Some special bin has no large item: All single large item bins have size much bigger than $1/2$ (actually $> 5/8$).
  Some small cases are tedious (and tight).

Tightness of the algorithm

Suppose that online algorithm ALG has ratio smaller than $5/3$.
Use this instance:

- 6 items of size $1/7$
  $\mathrm{OPT} = 1$, thus $\mathrm{ALG} = 1$.

Tightness of the algorithm

Suppose that online algorithm ALG has ratio smaller than $5/3$.
Use this instance:

- 6 items of size $1/7$
  $\mathrm{OPT} = 1$, thus $\mathrm{ALG} = 1$.
- 6 items of size $1/3 + \varepsilon$
  $\mathrm{OPT} = 3$, thus $\mathrm{ALG} = 4$.

Tightness of the algorithm

Suppose that online algorithm ALG has ratio smaller than $5/3$.
Use this instance:

- 6 items of size $1/7$
  $\text{OPT} = 1$, thus $\text{ALG} = 1$.
- 6 items of size $1/3 + \varepsilon$
  $\text{OPT} = 3$, thus $\text{ALG} = 4$.
- 6 items of size $1/2 + \varepsilon$
  $\text{OPT} = 6$, $\text{ALG} = 4 + 6 = 10$, a contradiction.

Bounded-space algorithms

- At most k bins are allowed to be open. A bin may be closed, i.e., later it cannot pack more items.

Bounded-space algorithms

- At most k bins are allowed to be open. A bin may be closed, i.e., later it cannot pack more items.

k-bounded-space BestFit [Csirik,Johnson,2001]

- The most full bin is closed when k bins are open and the next item does not fit into any of them.
- For $k \geq 2$, k-bounded-space BestFit has asymptotic approximation ratio 1.7.

Bounded-space algorithms

- At most k bins are allowed to be open. A bin may be closed, i.e., later it cannot pack more items.

k-bounded-space BestFit [Csirik,Johnson,2001]

- The most full bin is closed when k bins are open and the next item does not fit into any of them.
- For $k \geq 2$, k-bounded-space BestFit has asymptotic approximation ratio 1.7.

Optimal online algorithm [Lee,Lee,1985]

- There exists a family of k-bounded-space online algorithms whose asymptotic approximation ratio approaches $h_\infty \approx 1.691$ as $k \to \infty$.
- No k-bounded-space algorithm has a smaller asymptotic approximation ratio than $h_\infty \approx 1.691$.

Example – k-bounded space

- Input:

$$m \times \left(\tfrac{1}{7} + \varepsilon\right), \quad m \times \left(\tfrac{1}{3} + \varepsilon\right), \quad m \times \left(\tfrac{1}{2} + \varepsilon\right), \quad \text{where } m >> k$$

- k-bounded-space BestFit

$$\frac{m}{6} \times \left[\, 6\times \; \tfrac{1}{7} + \varepsilon \,\right] \qquad \frac{m}{2} \times \left[\, \tfrac{1}{3} + \varepsilon, \; \tfrac{1}{3} + \varepsilon \,\right] \qquad m \times \left[\, \tfrac{1}{2} + \varepsilon \,\right]$$

- Optimum:

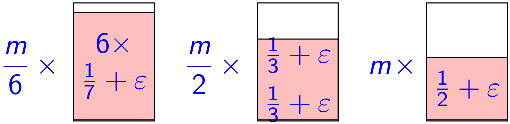
$$m \times \left[\, \tfrac{1}{7} + \varepsilon, \; \tfrac{1}{3} + \varepsilon, \; \tfrac{1}{2} + \varepsilon \,\right]$$

Example – k-bounded space

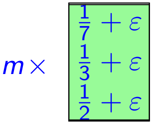
- Input:

$$m \times \left(\tfrac{1}{7} + \varepsilon\right), \quad m \times \left(\tfrac{1}{3} + \varepsilon\right), \quad m \times \left(\tfrac{1}{2} + \varepsilon\right), \quad \text{where } m >> k$$

- k-bounded-space BestFit or any k-bounded-space algorithm:



- Optimum:



Open problems

- Does the 1.7 absolute approximation ratio hold for **Almost Any Fit algorithms?** And for **3-Bounded Space Best Fit?**

- The best lower and upper bounds on the **asymptotic** approximation ratio are 1.54278 and 1.58889.