# On the price of clustering and related issues

**Leah Epstein**

University of Haifa, Haifa, Israel

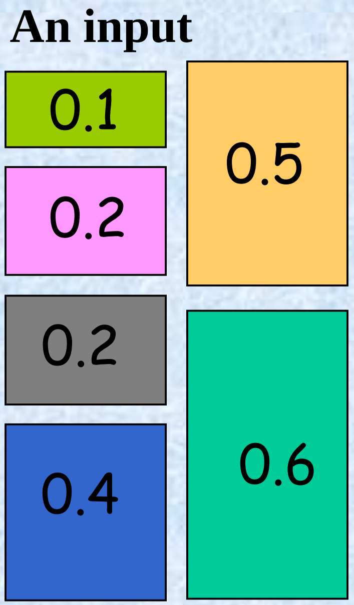
Online: receive items one by one

# Classic Bin Packing (BP)

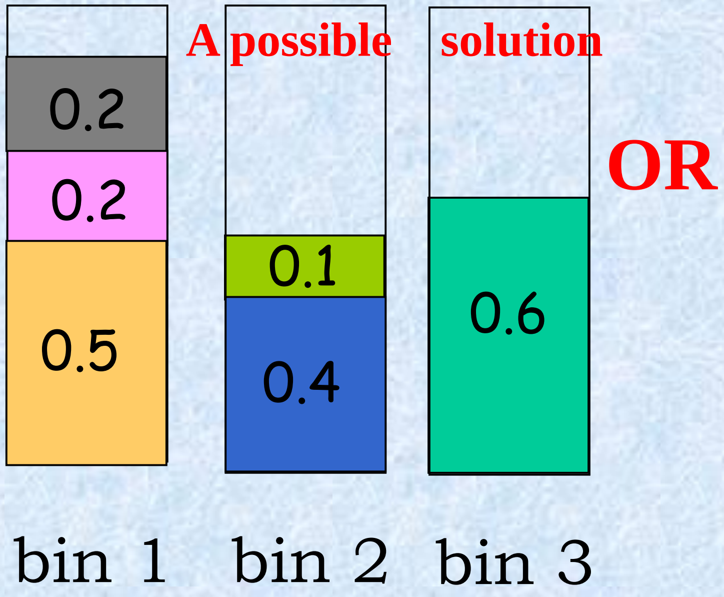
Offline: receive all items together

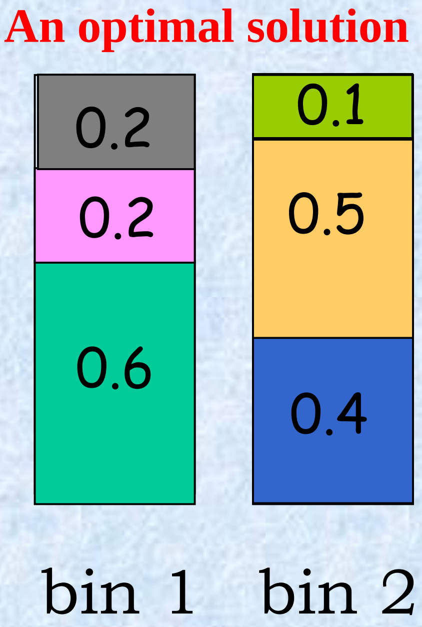
- A set of items. Item i has a size $s_i \in (0,1]$
- An unlimited supply of bins of capacity 1
- Pack into a minimum number of bins

Here, colors have no particular meaning

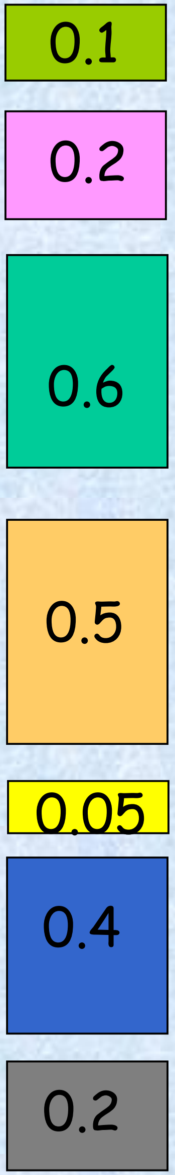


A small example





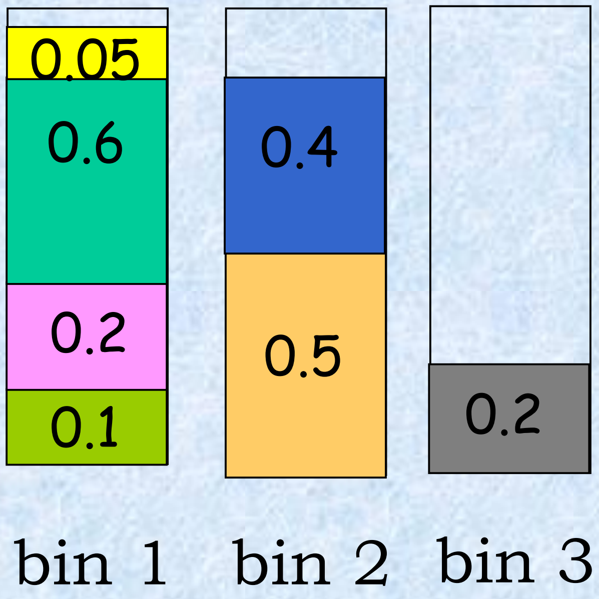
## Some greedy algorithms..



## First Fit (FF)

**Process the items as a list.**

**Pack each item into the minimum indexed bin where it can be packed**

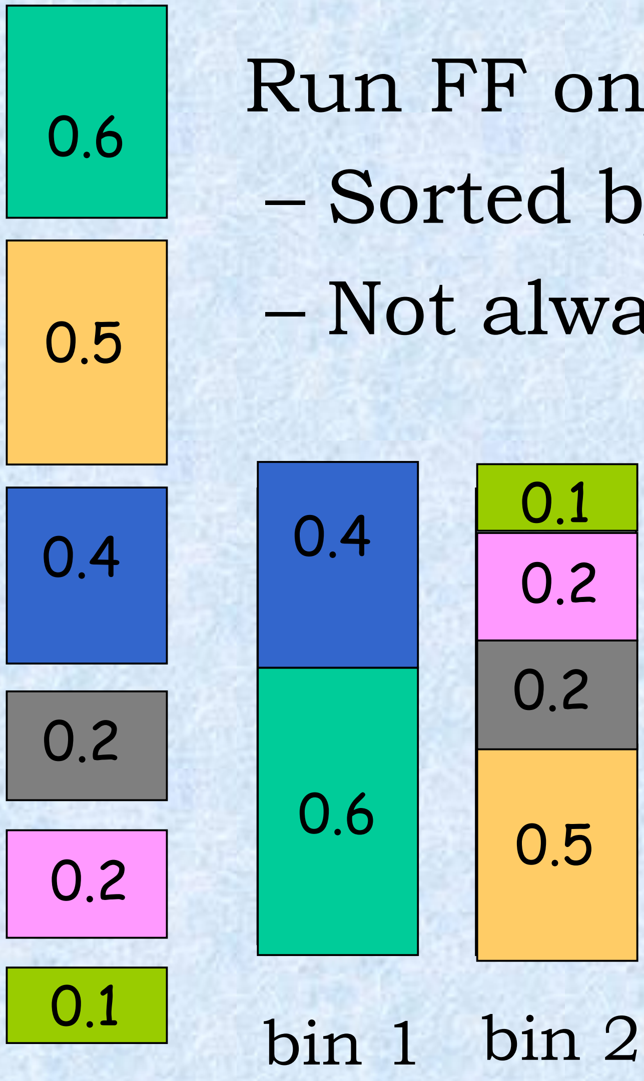


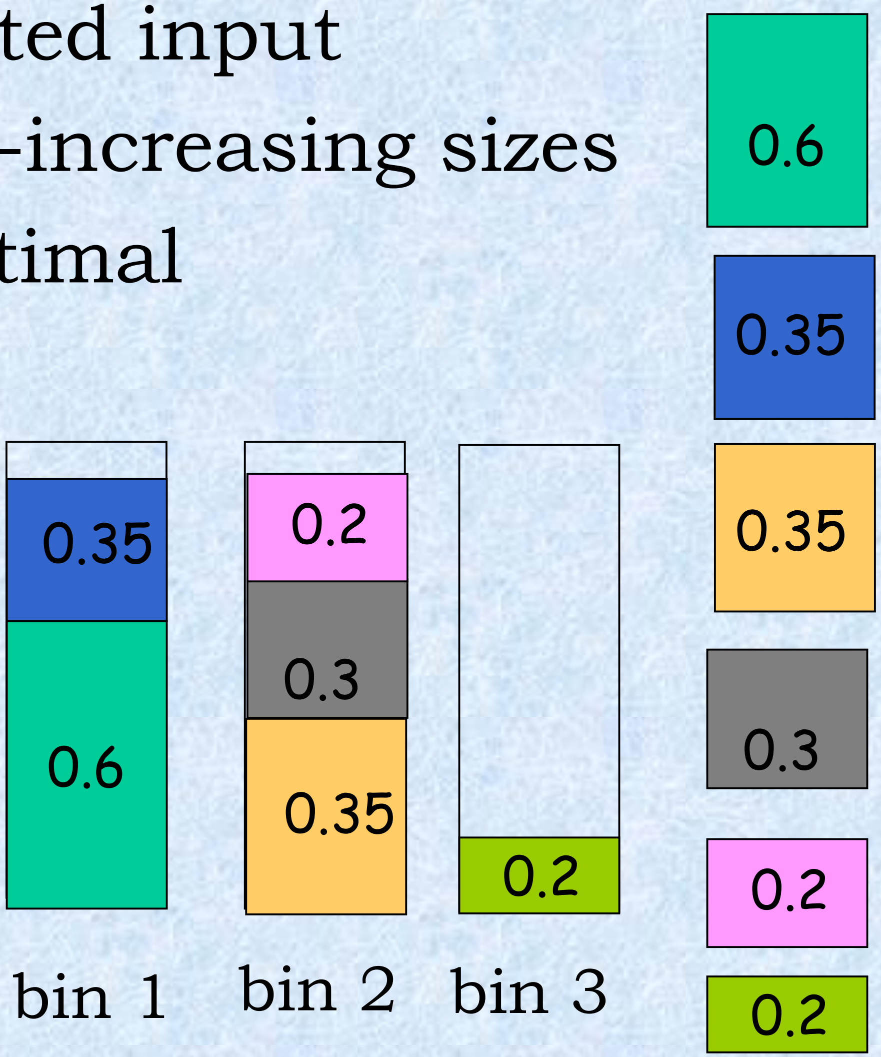
# First Fit Decreasing (FFD)

Run FF on a sorted input

– Sorted by non-increasing sizes

– Not always optimal



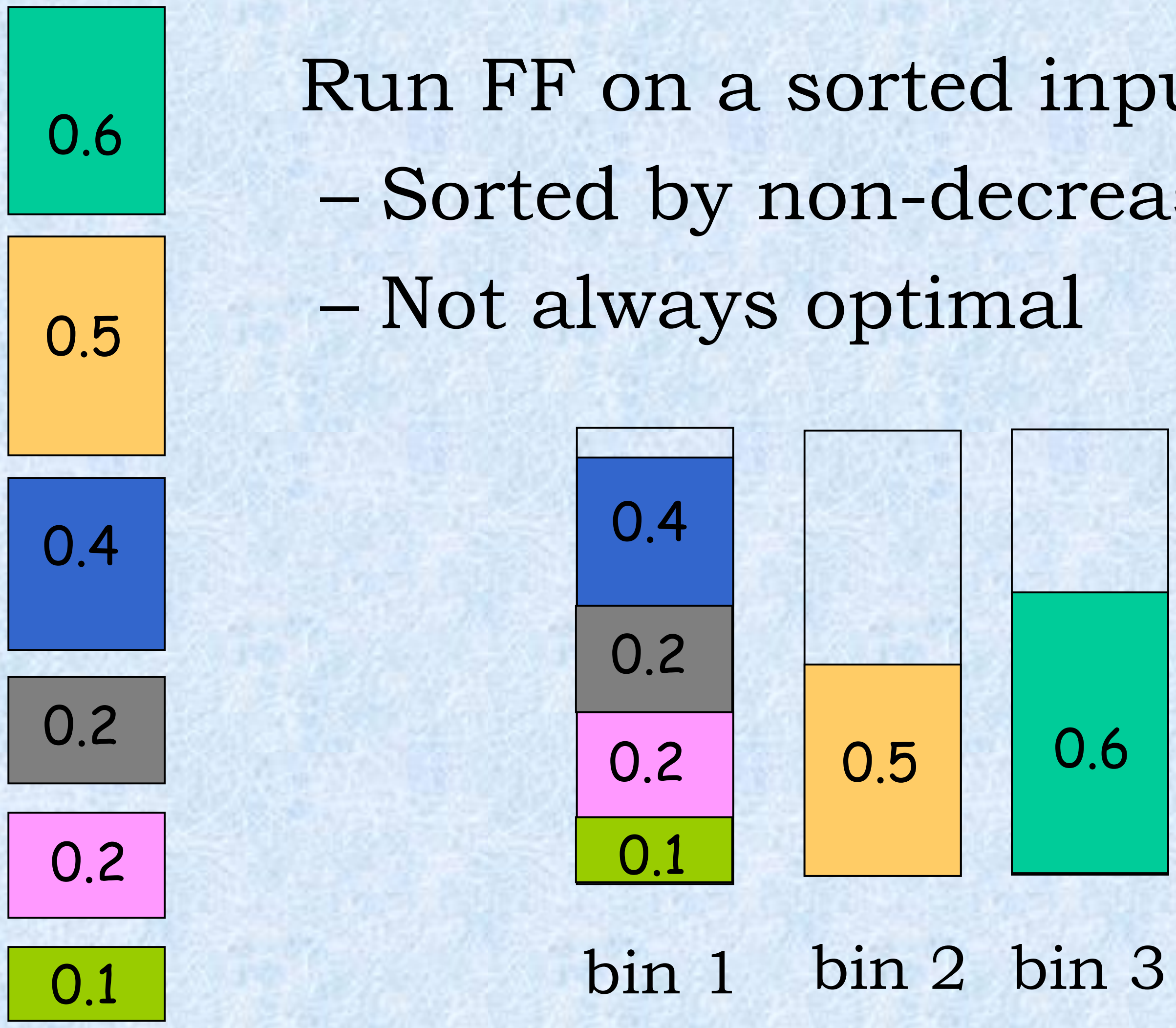


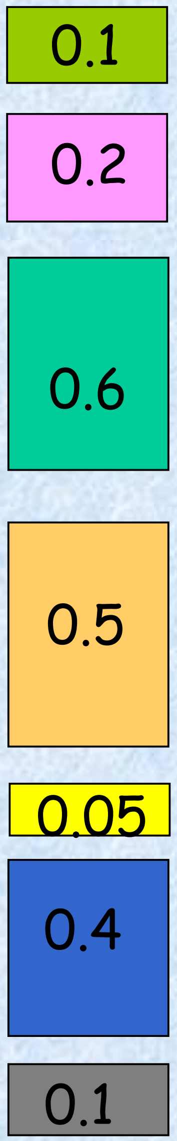
## First Fit Increasing (FFI)

Run FF on a sorted input

– Sorted by non-decreasing sizes

– Not always optimal



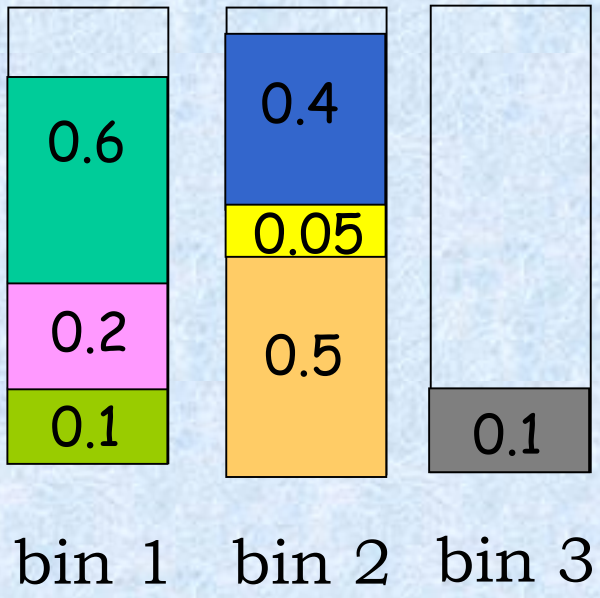


## Next Fit

Has one active bin

Packs an item into the active bin

If this is impossible, the active bin is closed and a new active bin is opened.



By sorting the items, one can apply **NFD** or **NFI**

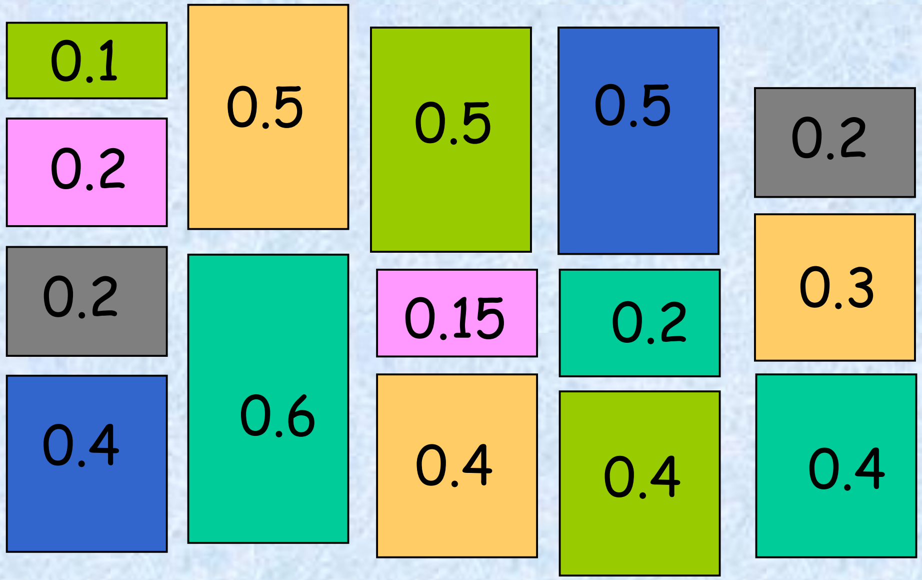
**NFI is the same algorithm as FFI for this problem**

## **Bin packing** with types

Every item has two attributes

A **size** **as always, and a** **type** (also called **color)**

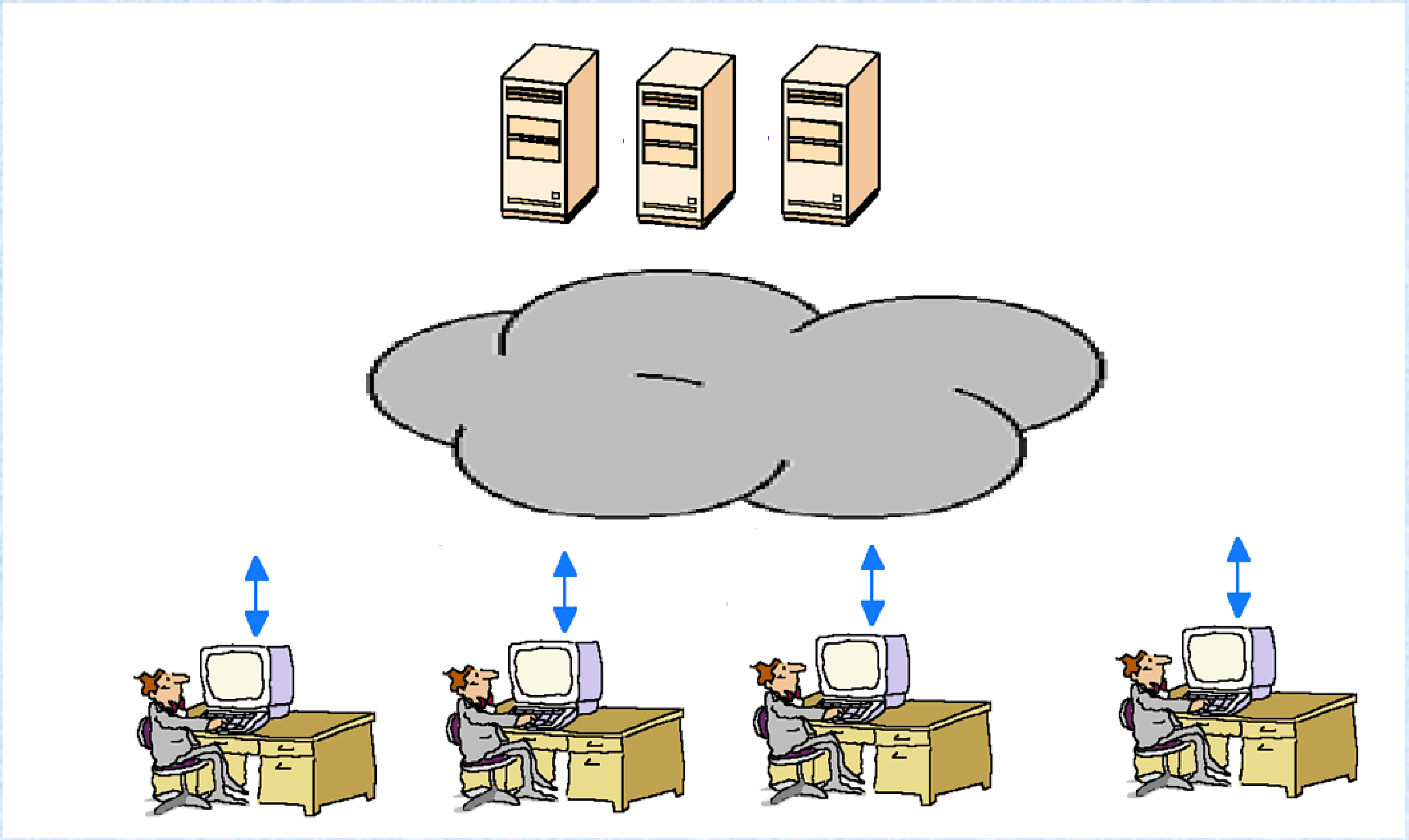
An example



**There are items of six types in the example**

**(Fake) applications in cloud computing: The types are locations of data**

## Motivation: Cloud Computing



## Clustered solutions

- These are solutions where **each bin can contain only items of one type**
- A **cluster** is the set of **items of one type**
- The packing problem is solved independently for every type/cluster
- **Every cluster may be packed optimally or using any other algorithm**
- **In an optimal clustered solution every cluster is packed optimally**
- **Application-wise: every cluster is a set of items of a user, which are placed in one location**

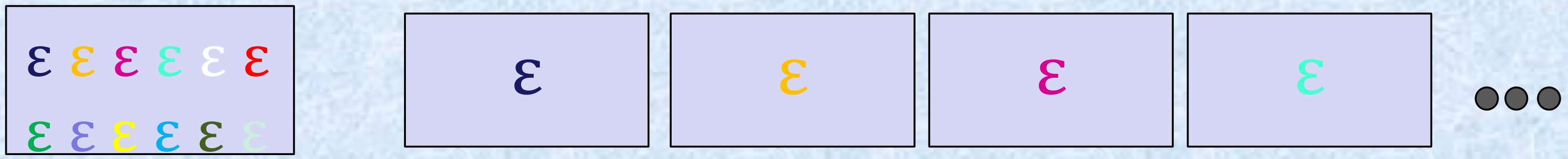
## Globally optimal solutions

- **These are solutions where the partition into clusters/types/colors is not taken into account**
- Such a solution is just a solution for standard bin packing, where items have sizes

  and this is the only attribute
- **All items are in one location**
- **This solution may be much better than an optimal clustered solution**

## Assumptions on clusters

- It is assumed that for every cluster, an optimal solution requires at least a given number of bins k
- **The parameter k is small: 2 or 3**
  - **Maybe 4..**
- **Why? Because otherwise it is possible that every item will be of a different type and items may be very small**



# Price of clustering

Introduced: [AESV'19]

Azar, Emek, van Stee, Vainstein, SPAA 2019

The price of clustering (PoC) is

the worst case ratio between

the optimal cost of a clustered solution and
the optimal cost of a globally optimal solution

The PoC is defined as a function of k

**k** = the lower bound on the optimal cost (number of bins) of a single cluster

As we saw, the PoC it is infinite if k=1

## The approximation ratio

**For minimization problems**

- **The strict measure**
  - Worst-case ratio between the cost of the algorithm and the cost of an optimal solution
  - Here "**the algorithm**" can **be a clustered solution,** an **optimal solution** is **an optimal global solution,**

  **and** the **approximation ratio is the PoC**

- **The asymptotic measure**

Compare only for inputs with sufficiently large optimal costs

Optimal cost of at least N, let N grow to infinity

**The results for the PoC will hold for both measures**

**Most other results are proved for the asymptotic measure**

**This is the standard measure for bin packing**

**Considered to be more meaningful for algorithms**

## Parametric bin packing

- In the standard bin packing problem, items can have **any size in (0,1]**
- In some applications, items sizes are **relatively small or bounded**
- The parametric variant with the parameter **b<1** is defined such that all items have sizes in **(0,b]**
- In some cases, it is assumed that $\boldsymbol{b = \frac{1}{t}}$ **for an integer t>1**

## Properties of parametric BP

For **FF and NF** and any output, it holds that every bin **except for possibly the last one has a total size of items above 1-b**

Thus, if b tends to zero, **the asymptotic approximation ratio tends to 1**

The absolute approximation ratio can be larger, for example $\boxed{\begin{array}{c}0.07\times 5\\ 0.08\times 8\end{array}}$ $\boxed{\begin{array}{c}0.05\times 12\\ 0.06\times 6\end{array}}$ $\boxed{\begin{array}{c}\\ 0.05\times 1\end{array}}$

Where the optimal solution is $\boxed{\begin{array}{c}0.06\times 6\\ 0.08\times 8\end{array}}$ $\boxed{\begin{array}{c}0.05\times 13\\ 0.07\times 5\end{array}}$

and such examples exist for smaller items as well

## The case k=2

- We will show why this case is not so interesting for classic bin packing,

  not even for the parametric case
- As we will see, the case **k=2** is **more interesting for other variants of bin packing**

**[AESV'19] the PoC for k=2 is 2**

Upper bound:

- **It is possible to analyze a greedy algorithm instead of an optimal solution**
- **In this analysis, the number of bins for each cluster is not smaller compared to an optimal solution for the cluster, and it is at least k**

# Upper bound – using FF

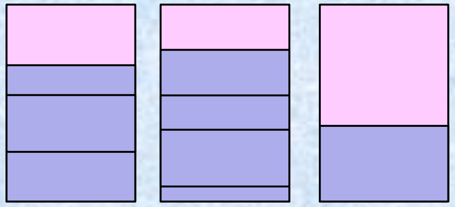
**Notation:**

$\boldsymbol{OPT}$ the number of bins for a globally optimal solution

$\boldsymbol{S}$ the total size of items $\boldsymbol{OPT \geq S}$

$\boldsymbol{OPT_i}$ the optimal number of bins for cluster i

$\boldsymbol{A_i}$ the number of bins used by FF for cluster i $\boldsymbol{A_i \geq OPT_i}$



at least two bins

**Bins of a given cluster i :**

**At least half full on average**

## Bins are at least half full:

because there are at least two bins

and no pair of bins can be combined

$$S \geq \frac{\sum A_i}{2} \geq \frac{\sum OPT_i}{2}$$

$$OPT \geq S$$

Almost trivial

$$\sum OPT_i \leq 2 \cdot OPT$$

*This trivial bound cannot be tight, can it?*

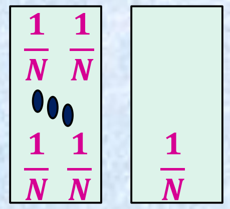
*or at least not tight for small items?*

*Can a cluster have items that do not fit into a bin, but they just barely do not fit, so the second bin is quite empty?*

## Tightness

For a large positive integer N

There are **N** clusters, where every cluster has $(N+1)$ **items**, **each of size** $\frac{1}{N}$



**and it requires two bins, for a total of 2N bins in the clustered solution**

**A globally optimal solution:**

**N+1 bins, every bin has N items**

**Ratio:** $\frac{2N}{N+1} \to 2$

**The tight ratio of 2 is valid also for any parametric case.**

**For any value of b, choose a sufficiently large N**

## Larger values of k: lower bounds

**The lower bound on the PoC for k=3 by [AESV'19] is ~1.93344**

**The greedy sequence for bin packing is used:**

$$\frac{1}{2}+\varepsilon,\ \frac{1}{3}+\varepsilon,\frac{1}{7}+\varepsilon,\ \frac{1}{43}+\varepsilon,\frac{1}{1807}+\varepsilon,\ldots$$

An **optimal solution** has one set of bins with these items and another set **of the same cardinality** N **(a large positive integer)** with **pairs of items** of the form $\frac{1}{2}+i\cdot\delta,\ \frac{1}{2}-i\cdot\delta$

**Where $\delta$ is much smaller than $\varepsilon$**

$\frac{1}{2}+\varepsilon$

$\frac{1}{3}+\varepsilon$

$\frac{1}{7}+\varepsilon$

$\frac{1}{43}+\varepsilon$

● ● ● but finite

$N$ bins

$\frac{1}{2}+i\cdot\delta$

$\frac{1}{2}-i\cdot\delta$

$i=1,2,\dots,N$

$N$ bins

$\delta\cdot N<\varepsilon$

## The clusters:

$\frac{1}{2}+(i+1)\cdot\delta$ $\frac{1}{2}+\varepsilon$

$\frac{1}{2}-i\cdot\delta$

Two additional items are added arbitrarily to some cluster

**N-1 clusters**
**no two items fit together and therefore three bins are needed for each cluster**

$\frac{1}{3}+\varepsilon$ Five times

**N/5 clusters**
**no three items fit together and therefore three bins are needed for each cluster**

$\frac{1}{7}+\varepsilon$
13 times

$\frac{1}{43}+\varepsilon$
85 times

**N/13 clusters** **N/85 clusters**

● ● ●

**A globally optimal solution**
**OPT with 2N bins**

$$\frac{3\left(N+\frac{N}{5}+\frac{N}{13}+\frac{N}{85}+\cdots\right)}{2N}\sim 1.93344$$

## Improvements, generalizations..

**E., 2019 a lower bound of 1.93558 for k=3**

One idea: the cluster

$\frac{1}{2}+\delta$

$\frac{1}{3}-\varepsilon$

$\frac{1}{6}+\delta$

$N/5$ bins

can be replaced with

$\frac{1}{3}+\varepsilon$ Five times

Four times $\frac{1}{3}+\varepsilon$ and one $\frac{1}{3}-\varepsilon$

**Still N/5 such clusters, still no three items fit together and therefore three bins are needed for each cluster**

**But some of the bins of OPT with such items can be different and there will be clusters with**

$\frac{1}{6}+\varepsilon$ 11 times

**Using items slightly smaller than 1/6 packed with some items of size $\frac{1}{3}+\varepsilon$ allows a further improvement**

# What about larger values of k?

**Similar constructions**

**for example, k=4**

$\frac{1}{3}+\varepsilon$ Seven times

**no three items fit together and therefore four bins are needed for each cluster**

$\frac{1}{7}+\varepsilon$ 19 times

**no seven items fit together and therefore four bins are needed for each cluster**

$\frac{1}{6}+\varepsilon$ 16 times

**no six items fit together and therefore four bins are needed for each cluster**

**Would this lead to the tight bound for k growing to infinity? Do we get the standard 1.69103? Yes, only the bins of OPT with**

$$\frac{1}{2}+\varepsilon,\ \frac{1}{3}+\varepsilon, \frac{1}{7}+\varepsilon,\ \frac{1}{43}+\varepsilon, \frac{1}{1807}+\varepsilon, \ldots$$ **are needed,**

**because the effect of one sparse bin per cluster becomes insignificant**

## Upper bounds

- The upper bound of [AESV'19] for k=3 on the PoC is based on defining linear constraints on the solution, and solving it using a computer
- The resulting upper bound on the PoC is 1.951
- It should have been 1.95, minor errors introduced due to the computer assisted proof and rounding
- This value was improved to
- 581/300~1.93667 E. 2019

  For simplicity, we will show how to obtain 1.95 easily
- We use weight functions that connect a globally optimal solution to an optimal clustered solution
- The total weight of all items is fixed for a given input and we bound it using both solutions

## An upper bound via a weight function

We use the next function: $w: [0, 1] \rightarrow [0, 1.95]$

**(for the improved upper bound, the weight function has five cases rather than two)**

$$w(x) = 1.8 \cdot x + \begin{cases} 0.15 & for \quad x > \frac{1}{2} \\ 0 & otherwise \end{cases}$$

The main task is that of finding a good weight function, the rest is easier

**Claim: For every bin (of a globally optimal solution), the total weight is at most 1.95**

**Proof: The first part of the weight function is multiplied by at most 1 for all items together, since their total size is at most 1. There is at most of item of size above ½ that might add 0.15.**

## An analysis of one cluster

To complete the proof, we show that for every cluster, **the total weight of its items is at least the number of bins for the cluster** in **an optimal solution for it**

Instead of an optimal solution, **we use FFD for the cluster** – so the number of bins it **still** **at least k.**

The **number of bins could increase** but this only means that we get a **larger lower bound on the total weight**

## Analysis for a cluster

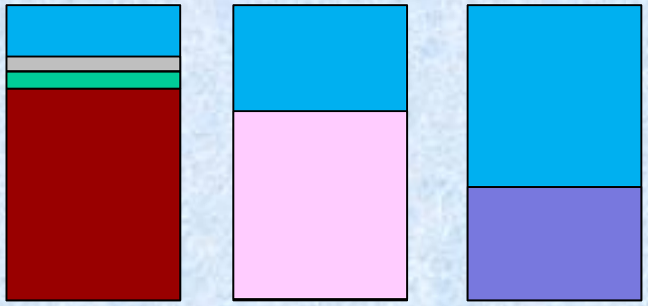
**There are two main cases in the proof.**

**We visualize several special cases of these cases, just to see the main idea.**

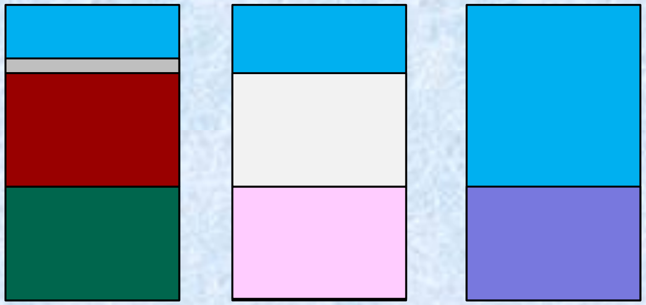
1. Three bins, there are two items of sizes above ½. **Those two items are packed into the first two bins by FFD.** Every two bins have total size above 1, and therefore **the total size for the three bins is above 1.5.**

**We have a total weight of at least:**

$1.8 \cdot 1.5 + 2 \cdot 0.15 = 3$



2. Three bins, there are no items of sizes above ½. The first item of the last bin has size in (1/3,1/2]. **The items of sizes in (1/3,1/2] are packed first**, and therefore there are at least five such items. **We have a total weight of at least:** $\mathbf{1.8 \cdot \left(\frac{5}{3}\right) = 3}$



3. Three bins, there are no items of sizes above ½. The first item of the last bin has size in (0,1/3]. The other two bins have total sizes above 2/3. Every pair of bins has total size above 1 together.

**We have a total weight of at least:** $\mathbf{1.8 \cdot \left(\frac{2}{3} + 1\right) = 3}$

## The modified weight function

$$w(x) = \frac{21}{13} \cdot x + \begin{cases} \frac{997}{3900} & for \quad x > \frac{1}{2} \\ \frac{256}{3900} & for \quad \frac{1}{3} < x \le \frac{1}{2} \\ \frac{216}{3900} & for \quad \frac{1}{4} < x \le \frac{1}{3} \\ \frac{40}{3900} & for \quad \frac{1}{6} < x \le \frac{1}{4} \\ 0 & for \quad x \le \frac{1}{6} \end{cases}$$

## Batched bin packing

For a parameter $q>1$, items are presented in q batches

- All items of a batch appear at once
- The items of a batch should be packed before another batch is presented

(q+1) batches is never easier than q since batches could be empty

<u>Two variants</u>

1. Use the same bins for all batches
   - Open new bins, if necessary, at any time
   - This is an intermediate model between online and offline
2. Must use separate bins for different batches
   - This variant can be seen as an offline problem
   - It is related to clustered solutions

and we will only discuss this second batched model

## Does the separate bins model lead to an approximation ratio of q?

- **For the absolute approximation ratio**

  **YES, it does.**

  **An algorithm that packs each batch optimally separately has at most this ratio**

  **Even by splitting an optimal solution into the separate batches, every bin is split into at most k bins**

- **If there are q items of sizes 1/q**
  - **Each appears in its own batch**
  - **An optimal solution for the entire input uses one bin, but any batched solution has q bins**
- **For the asymptotic approximation ratio?**
  - **Not at all!**

## Simple bounds for two batches

**What about the following simple approach, which has to be the best possible:**

- **Pack the first batch optimally**
- **Pack the second batch optimally into new bins**

An example (lower bound) for an integer N:

**Batch 1: 2N+1 items, each of size 0.4**

**Batch 2: 2N+1 items, each of size 0.6**

**OPT for the entire input:** 2N+1 bins **with {0.4,0.6}**

**An optimal solution for batch 1:**

**N times {0.4, 0.4}, one {0.4}**

**An optimal solution for batch 2:**

**2N+1 times {0.6}**

**(N+1)+(2N+1)=3N+2 bins versus 2N+1 bins**

# Known upper bounds

- FFD has asymptotic approximation ratio for two batches 19/12~1.58333 [**Dósa2017]**
- Why is that?
  - q=2 is much harder than offline
  - Since FFD is applied, the solutions

  for the batches are not always optimal
  - Split an example into two parts

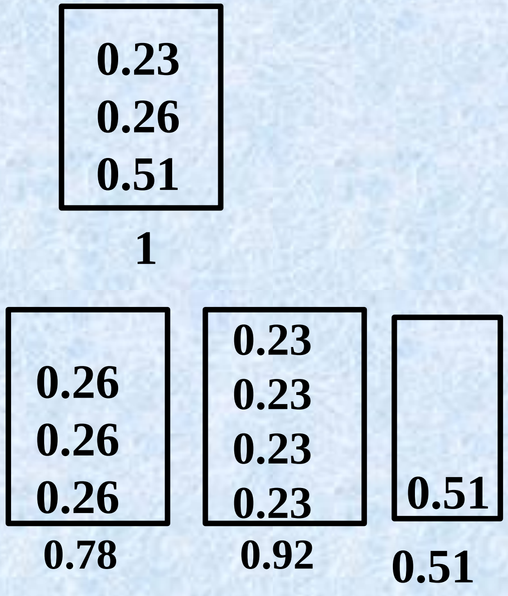
  First batch: **12N items of sizes 0.26**

  and **12N items of sizes 0.23** (7N bins)

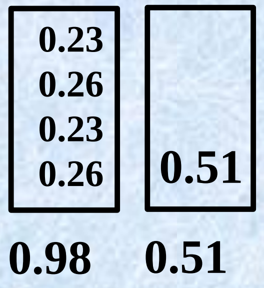
  In a second batch, **12N items of sizes 0.51**

  12N new bins

  **OPT=12N, ALG=19N**



**Our algorithm from the previous slide will create 18N bins**



## The separate bins model does not lead to a high **asymptotic approximation ratio**

- The ratio is always (much) smaller than 2
- Even for large q, the value ~1.69103 is tight
- For two batches the ratio is 1.5
  - 5/3 for three batches
  - Etc.
- **For simplicity we will solve NP-hard problems to optimality**
  - Online algorithms are not limited to certain running times
  - It is possible to obtain almost the same bound by using an asymptotic approximation scheme instead
    - The additive constant may be larger

## The optimal algorithm for two batches (E., 2016)

Our algorithm has an absolute approximation ratio of 2 and an asymptotic approximation ratio of 3/2

In our example, we found a family of instances for all odd values of ALG where

$$OPT = 1.5 \cdot ALG + 0.5$$

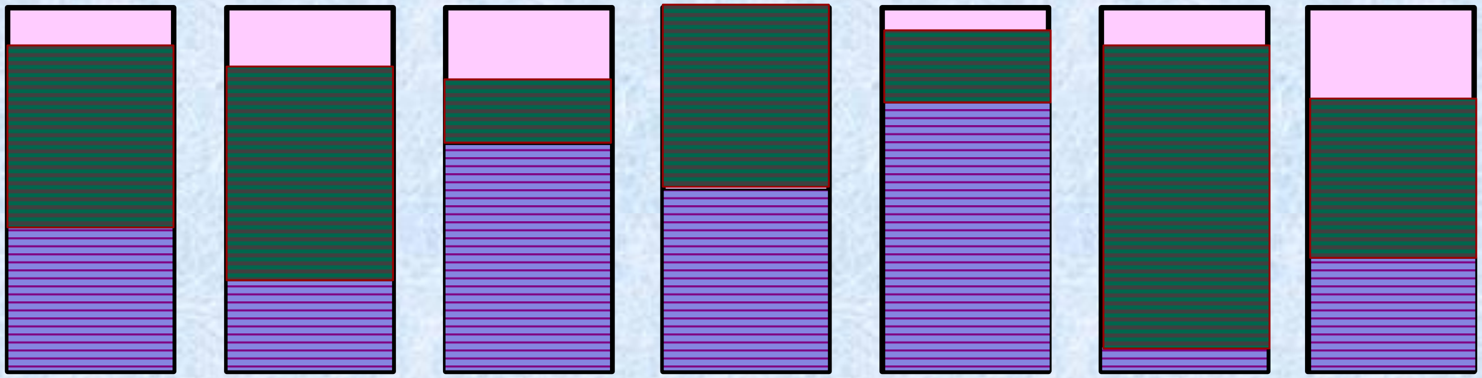
Where ALG denotes the total cost for the algorithm, that is, for both batches together.

The example shows that this is the tight absolute/asymptotic approximation ratio of the algorithm for separate bins,

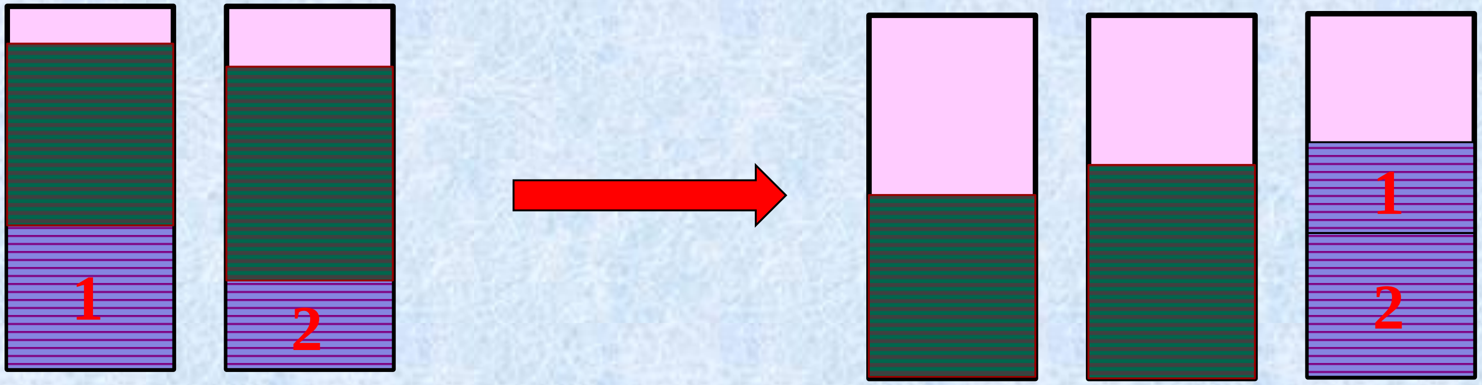
using **ALG=1, ALG**$\rightarrow \infty$

## Analysis

- Consider an optimal solution for the entire input (both batches)
  - with X bins
- We construct certain solutions for the two batches separately
  - **Not necessarily optimal ones**
  - **We can consider their total cost**
  - **The sum of costs of optimal solutions for the two batches is not larger**



An optimal solution



Creating three bins out of two bins such that items of different batches are packed separately

## Easy, but is this really helpful?

- **Every two bins become three**

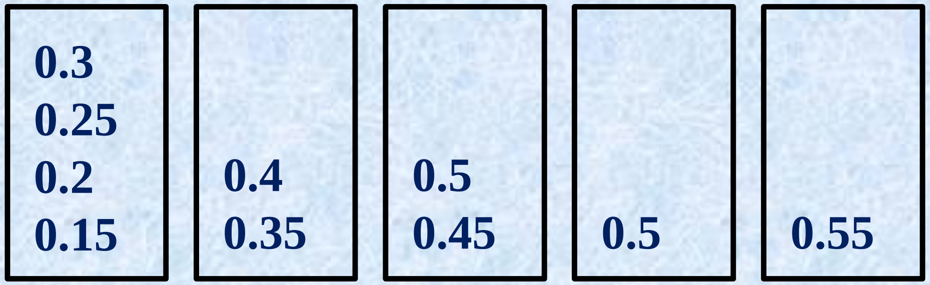
  Either all the items of these pair of bins for the first batch have total size **at most 1**, or else all the items those of the second batch have total size of **at most 1** **(it is possible that both properties hold)**
- **If there is an odd number of bins in OPT, the last bin is converted into two bins**
- We got at most 1.5·X+0.5 bins from X bins (we get just 1.5·X bins if X is even)

## But doesn't the ratio get much larger for large q, even if we pack each batch optimally?

- For the general case, we define **"mega-items"** and define a solution for them
  - **Packing each batch separately**
- This is done offline, just for the proof
  - **It is not an algorithm!**
- A **mega-item** is a maximal subset of items packed into one bin of OPT and belonging to one batch
- We already defined this for q=2
  - **every bin of OPT had at most two mega-items**

## How to pack mega-items?

- We can use FFI (First Fit Increasing)
- Why? It is equivalent to NFI
  - **NF: Use one active bin for packing**
  - **Open a new active bin when cannot pack**
  - **Never return to a bin that could not be used**
- An example:
  0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5,0.5,0.55



**But this is not optimal!**

## Analysis

- The packing is indeed not optimal, but in this way we can analyze all batches together rather than separately
- We can use a weight function
  - [Baker, Coffman 1981]
- The function as used for NFI for actual items (not mega-items)
  - Actually, it was used for NFD, but they have the same number of output bins for every input [Fisher'88]

## But…

- There is an additive constant for every batch
- So we get that the cost is ~1.69103

times **OPT plus O(q)**

**Recall that the value 1.69103 is a sum of a series 1+1/2+1/6+1/42+1/1806+…**

- **The ratio is smaller for fixed q**
  - **The sum of the first q values**
  - **3/2 for q=2**
  - **5/3 for q=3**

## Still on drawbacks

- Can we avoid such a large additive constant? No
- because of the lower bound of q on the absolute approximation ratio with separate bins
- What about the case where bins are not separate for the separate batches?
  - Harder to analyze
  - The bas examples are the same
  - Already for q=3 a purely online algorithm is better than the one we could analyze where bins are separate

new

## Open end bin packing problems (OEBP)

In such problems a bin can have **items of total size above 1** but the total size **after removing an item has to be strictly below 1**

**Max-OEBP:** the total size has to be below 1 after excluding **the largest item** **(or excluding any item)**

**Min-OEBP:** the total size has to be below 1 after excluding **the smallest item**

**one item "sticks out"**

**Why study this?**

**Because it is similar to bin packing,**

**yet it is quite different**

## Examples and differences

**{0.3,0.4,0.5}** the total size is **1.2**

but the bin is valid for both variants

|  |  |  |
|---|---|---|
| Max-OEBP | {0.3,0.4} | 0.7 |
| Min-OEBP | {0.4,0.5} | 0.9 |

**{0.2,0.3,0.4,0.5}** **1.4**

|  |  |  |  |
|---|---|---|---|
| Max-OEBP | {0.2,0.3,0.4} | 0.9 | **Valid** |
| Min-OEBP | {0.3,0.4,0.5} | 1.2 | **Not valid** |

# Some known results for OEBP

**For standard BP we will not discuss such results since some of us are familiar with them.**

There are two other variants of OEBP, which we will not discuss either.

**All variants of OEBP have asymptotic polynomial time approximation schemes**

**For the two problems studied here:**

**Leung, Dror, Young, 2001, Lin, Yang, Xu, 2006, 2010**

**Lin, Yang & Xu also analyze greedy algorithms and online algorithms for min-OEBP, in particular for the parametric case where 1/b is an integer**

**Other results for online algorithms are proved or can be deduced from Yang and Leung, 2003, E. & Levin, 2020, Balogh, E., Levin, 2020.**

## Greedy algorithms for Max-OEBP

FF for this variant: **add every new item to the bin of the smallest index that will remain valid after the addition**. For example, {0.2,0.7,0.2} is valid.

It is possible to add 0.4 but impossible to add 0.65, since 0.8<1 but 1.05>1

**A bin of total size below 1 can always receive another item, so every bin (except for the last one) will have total size of at least 1.**

**This is not true to Min-OEBP, the bin {0.2,0.2} with respect to an item of size 0.9.**

**For Min-OEBP the property is nevertheless true for FFD because the every new item is the current smallest item.**

**The only greedy algorithm studied previously for Max-OEBP is FFD (Ongkunaruk,2005).**

She showed that if all items have sizes in [0,1), the asymptotic approximation ratio is 1.5.

**We can in fact show this even if there may be items of size 1, though the proof requires a new ingredient.**

**The reason for the slightly longer proof is that as long as only items of size 1 are packed (first), each one is packed into its own bin, and items of sizes close to 1 can be added later.**

**Here, we will discuss FF for Max-OEBP.**

The general case is not very interesting: The upper bound is 2 since no bin can have a total size of 2 or more, and almost every bin of FF has a total size of 1 or more. The lower bound example is also simple (next slide).

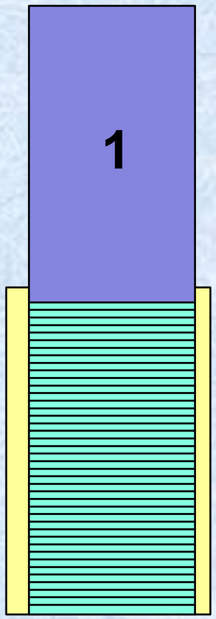
OPT has M bins with $(N-1) \times \frac{1}{N}, 1$

If FF gets the small items first, they are packed into bins of the form $N \times \frac{1}{N}$ that cannot receive additional items.

Afterwards, it has M bins of the form $\{1\}$

In total, $M \cdot \frac{N-1}{N} + M$ bins,

and the ratio tends to 2 for large N.



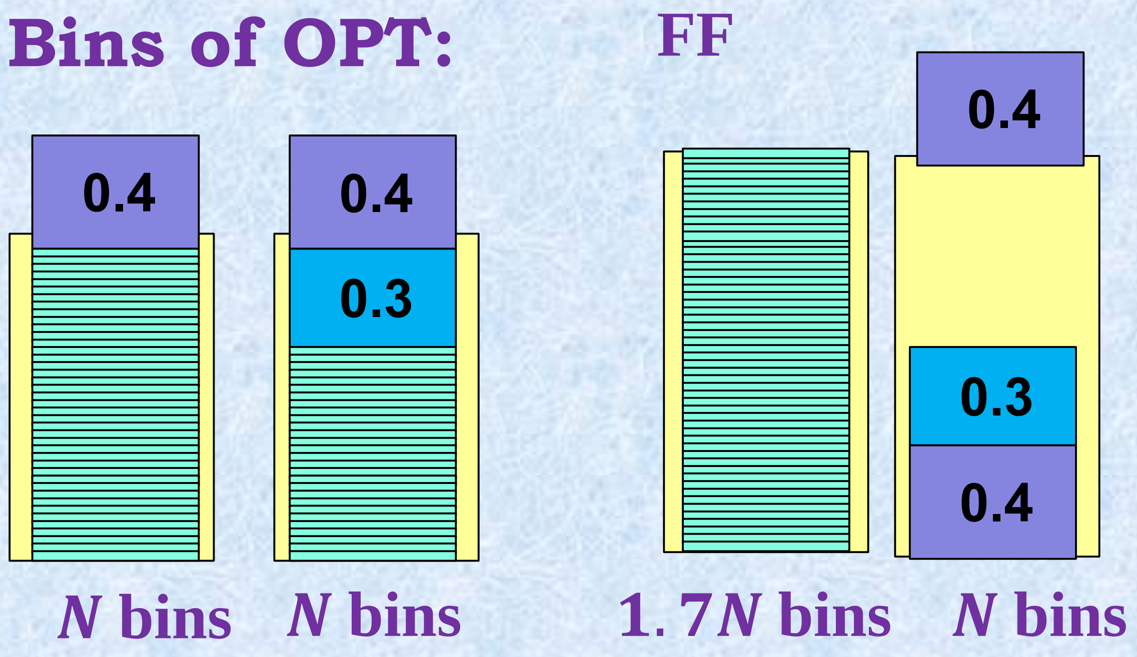
**Can we avoid items of size 1?**

- **For b=0.999999 the asymptotic approximation ratio is 1.5**
- **The same ratio holds also in the case where b=1 but items of size 1 are impossible, because no bin of FF can have just one item.**

## An example for the parametric case

The asymptotic approximation ratio never exceeds 1+b, but in some cases it is smaller.

Consider the case b=0.4



The resulting ratio is 1.35 and it is tight. The proof requires a new weight function and a long calculation. A class of weight functions is defined such that all values of b are covered by the proof

## The weight function

For the case $b \in (\frac{1}{3}, \frac{1}{2}]$

$$w(x) = \begin{cases} x & for \quad x < \frac{1}{3} \\ \frac{x+1}{4} & for \quad \frac{1}{3} \le x \le b \end{cases}$$

For the case $b \in (\frac{1}{4}, \frac{1}{3}]$

$$w(x) = \begin{cases} x & for \quad x < \frac{1}{4} \\ \frac{2x+3}{3} & for \quad \frac{1}{4} \le x < b \end{cases}$$

Etc.

## Batched bin packing for Max-OEBP

**Here, the asymptotic approximation ratio is 2 for any number of batches q≥2**

**The lower bound once again consists of items of size 1 and items of size 1/N, such that by splitting them into two batches the result is similar to the output we saw for FF**

**The upper bound is also standard, since bins are half full on average. The additive constant depends on q and this cannot be avoided**

# The PoC for Max-OEBP

**The value of the PoC is 3 if every cluster requires at least two bins (k=2).**

**In the lower bound example, there are items of sizes 1 and items of size $\varepsilon$**

**An optimal solution has N bins with**

$\{\varepsilon, \varepsilon, \ldots, \varepsilon, 1\}$ Total size below 1 for the epsilons

**There is a cluster with all items of size 1, which requires N bins**

Total size $1 + 2\varepsilon$

**Other clusters are of the form $\{\varepsilon, \varepsilon, \ldots, \varepsilon\}$, so every cluster requires two bins, which adds almost 2N bins.**

## The PoC for Min-OEBP with k=2

**The value of the PoC in the case k=2 is 4** **(every cluster requires at least two bins)**

**In the lower bound example, there are items of sizes $1-\varepsilon$ and items of size $\varepsilon$**

**An optimal solution has N bins with**

**$\{1-\varepsilon\ ,\ 1-\varepsilon\ \}$ and one bin with $\{2\varepsilon, 2\varepsilon, \ldots, 2\varepsilon\}$**

**Clusters are of the form $\{1-\varepsilon, 2\varepsilon, 2\varepsilon\}$.**

Total size of 1

**In this variant, often it is the case that every bin of an optimal solution has identical or very similar items**

**There are 2N clusters. For every cluster, the sum without the smallest item the sum is $1+\varepsilon$ , so every cluster requires two bins, and there are 4N bins in total.**

## An example for the parametric case

Here, we unfortunately do not have tight bounds for all values of b. But we do have such bounds for some cases.

**For example, in the case b=0.8, we can show tight bounds of 3**

The upper bound requires yet another complicated weight function.

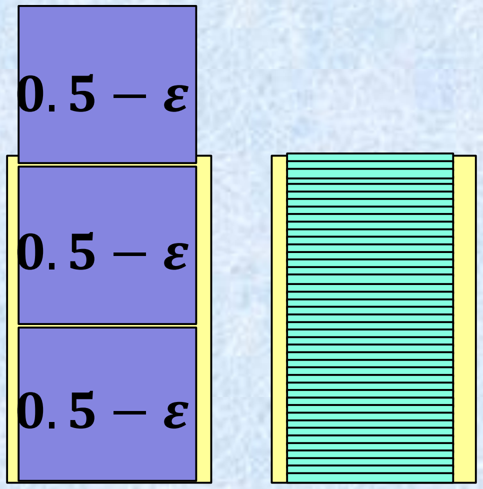
There are two different lower bound constructions because the situation is different in the two cases [0.5,0.8], [0.8,1)

# Two lower bound constructions for b=0.8 and the PoC for Min-OEBP with k=2

OPT

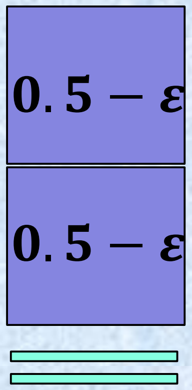
Items are much smaller than 0.8

Every cluster



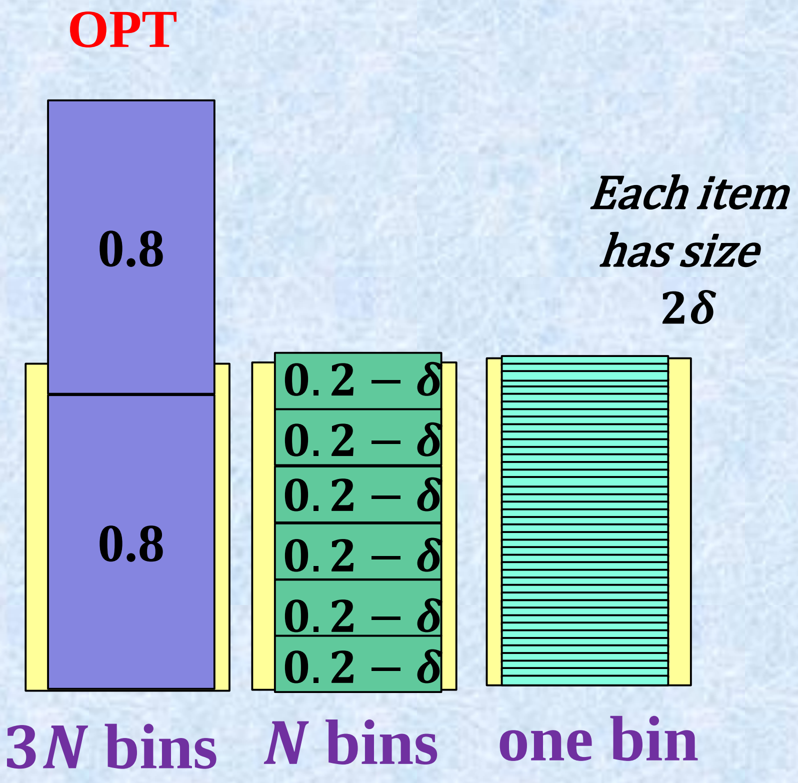
$N$ bins one bin

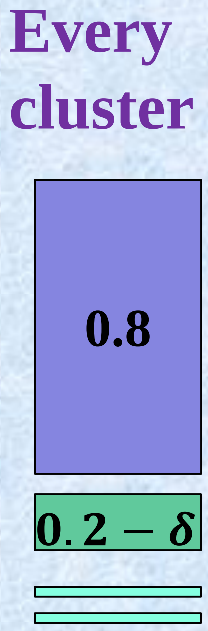
*Each item has size* $3\varepsilon$



There are $1.5\cdot N$ clusters, and each one requires two bins
There are $3\cdot$ N bins

# Two lower bound constructions for b=0.8 and the PoC for Min-OEBP





There are $6\cdot N$ clusters, and each one requires two bins
There are $12\cdot$ N bins

## The weight function for b=0.8

$$w(x) = \begin{cases} \frac{5}{4}\cdot x & for \quad x \le \frac{1}{5} \\ \frac{1}{4} & for \quad \frac{1}{5} \le x \le \frac{1}{4} \\ x & for \quad \frac{1}{4} \le x \le \frac{3}{4} \\ \frac{3}{4} & for \quad \frac{3}{4} \le x \le \frac{4}{5} \end{cases}$$

The function w is continuous again.

The point ½ is not a breakpoint of the weight function, but it does have a special role in the proof

## Thank you!

for connecting to Zoom
and for listening

Questions?
Comments?
Complaints?
Jokes?
New results?