Bin packing:

Input: Set of items I with sizes in (0,1].

Goal: Pack all items into a minimum number of bins of unit capacity.

- Known to be NP-hard [Reduction from PARTITION]

Online Bin Packing:

- Items arrive one-by-one.
- They need to be packed irrevocably, without knowledge of the future.

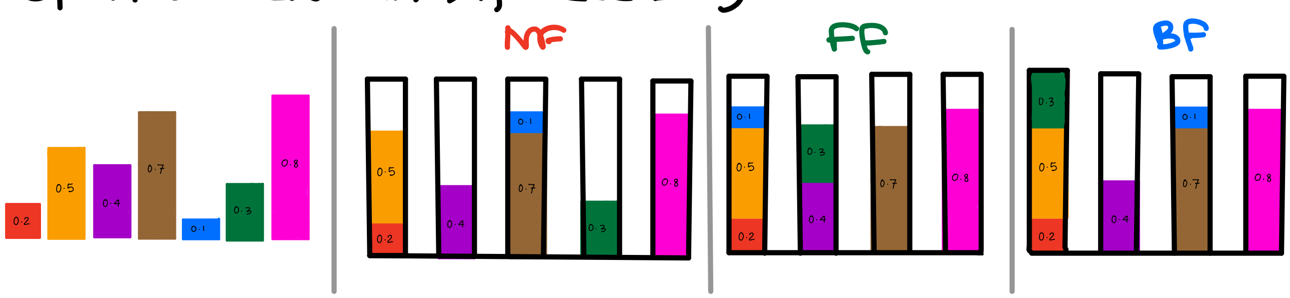
Pack the incoming item into

Next-Fit (NF): the bin opened most recently, if it fits.

First-Fit (FF): the first opened bin where it fits.

Best-Fit (BF): the fullest bin, where it fits.

Open a new bin, if necessary



- Competitive Analysis of Next Fit:

Say NF uses bins $B_1, B_2, \dots, B_m$.

Key property:

we opened a new bin

$$size(B_i) + size(B_{i+1}) > 1 \quad \forall i \in [m-1]$$

Now, $$OPT \geq size(I) = \sum_{i=1}^{m} size(B_i)$$
$$= \frac{1}{2}\left[\sum_{i=1}^{m-1} (size(B_i) + size(B_{i+1}))\right]$$
$$+ \frac{1}{2}\left[size(B_1) + size(B_m)\right]$$
$$> \frac{1}{2} \cdot (m-1)$$

$\Rightarrow m < 2OPT + 1.$

$\Rightarrow m \leq 2OPT.$ (As m is an integer)

- It is tight!

Consider sequence $\frac{1}{2}, \epsilon, \frac{1}{2}, \epsilon, \dots$ (n items).

Then $OPT = n/4 + 1$. say, $\epsilon \leq \frac{2}{n}$.

$NF = n/2$.

- Any Fit Algorithm:

Open a new bin only when the newly arrived item fits in none of the prev. bins

- Almost Any Fit:

Any fit algo which avoids worst-fit strategy (avoid putting item in the least full bin)

- One can show almost any fit (includes BF/FF) algorithms have C.R. 1.7.

- Lower Bound for Bin Packing:

Input sequence:

$$(\underbrace{\tfrac{1}{2}-\epsilon, \tfrac{1}{2}-\epsilon, \dots, \tfrac{1}{2}-\epsilon}_{\substack{m \text{ items} \\ (I_1)}}, \underbrace{\tfrac{1}{2}+\epsilon, \tfrac{1}{2}+\epsilon, \dots, \tfrac{1}{2}+\epsilon}_{\substack{m \text{ items} \\ (I_2)}})$$

For I_1 : $OPT(I_1) = m/2$.

Say, $ALG(I_1) = \alpha m$, $\frac{1}{2} \leq \alpha \leq 1$.

Then, C.R. $\geq \dfrac{\alpha m}{m/2} = 2\alpha$.

For $I_1 \cup I_2$: $OPT(I_1 \cup I_2) = m$.

Let x, y be the number of 1-bins & 2-bins in packing of I_1 by ALGO.

Then $x + 2y = m$ (# items)

$x + y = \alpha m$ (# bins)

$\Rightarrow y = m - \alpha m$. $x = \alpha m - y = 2\alpha m - m$.

Now items in I_2 can not go into these y bins.
At max x of them can be packed in x 1-bins.

Remaining $m - x$ will require new bin.

Hence, $ALGO(I_1 \cup I_2) = \alpha m + m - x$

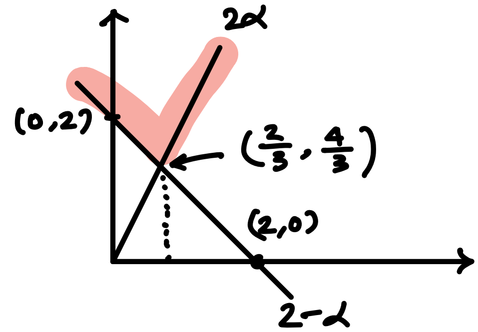
$= \alpha m + m - 2\alpha m + m = 2m - \alpha m$.

Hence, C.R. $\geq \frac{m(2-\alpha)}{m} = 2-\alpha$.

So, worst case C.R.

$\geq \max\{2\alpha, 2-\alpha\}$

$\geq 4/3$ for $\alpha = 2/3$.



• One can show a better lower bound of $3/2$ using sequence: m items of size $1/6 - \epsilon$,
m " " " $1/3 - \epsilon$,
m " " " $1/2 + 2\epsilon$.

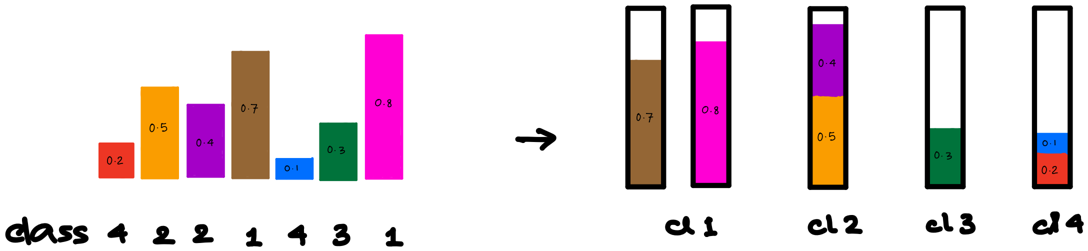
• Best Online Algorithm for Bin Packing.
(under $O(1)$ number of open bins)

Harmonic Algorithm: (H_K) (Lee & Lee, JACM '85).

→ create K classes:

$$(\tfrac{1}{2}, 1], (\tfrac{1}{3}, \tfrac{1}{2}], (\tfrac{1}{4}, \tfrac{1}{3}], \ldots, (\tfrac{1}{K}, \tfrac{1}{K-1}], (0, \tfrac{1}{K}].$$

→ Place members of each class separately.



Say, $K = 4$.

Analysis using Weighting Technique.

Weighting technique is a general technique

Step 1. Define a weight function $w(x)$ for item size x.

[Generally, $w(x) \geq x$ → also called 'rounding up'.

Step 2. Prove that any bin of ALGO has wt ≥ 1.

[except possibly a constant number of bins]

Step 3. Prove that maximum possible weight that can be put into a bin is $\leq J$.

- This will imply C.R. $\leq J$.

<u>Proof</u>:

step 2 $\Rightarrow$ $$\sum_{i \in I} w_i = \sum_{j \in [m]} \sum_{\substack{i \text{ is packed} \\ \text{in bin } B_j \\ \text{by ALGO}}} w_i \geq m =: \text{ALGO}. \quad \text{— (A)}$$

$B_1, \ldots, B_m$ denotes bins returned by ALGO.

step 3 $\Rightarrow$ $$\sum_{i \in I} w_i = \sum_{j \in [m']} \sum_{\substack{i \text{ is packed} \\ \text{in bin } B'_j \\ \text{by OPT}}} w_i \leq m' \cdot J = \text{OPT} \cdot J \quad \text{— (B)}$$

$B'_1, \ldots, B'_m$ denotes bins returned by OPT

(A) + (B) $\Rightarrow$ ALGO $\leq J \cdot$ OPT.

i.e. C.R. is J.

- An alternate way of seeing this is via primal dual.

LP Relaxation for bin packing:

$$\min \sum_{C\in\mathcal{C}} x_C : \quad \sum_{C\ni i} x_C \geq 1 \quad \forall\, i\in I, \quad x_C \geq 0.$$

$\mathcal{C}$ is set of all possible packing of a bin.
If a particular packing $C\in\mathcal{C}$ is selected, then $x_C = 1$, else $x_C = 0$.

Obj: min no. of selected bins.

constraint: Each item $i\in I$ must be packed.
[One of the C containg i is selected]

Say, OPT of this LP is P^*.

Dual of this LP:

$$\max \sum_{i\in I} v_i : \quad \sum_{i\in C} v_i \leq 1 \quad \forall\, C\in\mathcal{C}, \quad v_i \geq 0.$$

Say, OPT of this dual LP is D^*.

Then, $D^* = P^* \leq \text{OPT} = \text{integral OPT}$

LP Duality   LP Relaxation

Take $v_i = \frac{w_i}{J}$ then $\sum_{i\in C} v_i \leq 1$ (From step 3)

$$\text{OPT} \geq \sum v_i = \sum_{i\in I} w_i / J \geq \text{ALGO}/J$$

LP duality as v_i is a feasible dual soln

So weight functions can be thought of as a dual variable corrs. to each primal constraints.

- Creative part is in choosing the right wt function based on the property of algorithm.

- Analysis of Harmonic:

- Weight of an item in class i is $1/i$ when $i<K$.
- Weight of an item of size x in class K is $\frac{K}{K-1} \cdot x$.

Except possibly K "open" bins,

For other bins of type $i<K$, they have i items inside $\Rightarrow wt(Bin) \geq \frac{1}{i} \cdot i = 1$.

For bin of type K, it is at least full upto $> \frac{K-1}{K}$

$\Rightarrow wt(Bin) > \frac{K}{K-1} \cdot \frac{K-1}{K} = 1$.

Hence, step 2 is done.

To find the maximum total weight of item in a bin of OPT:

Define density of item of size x: $\frac{w(x)}{x}$.

To get the maximum profit, use a greedy algo that places items in nonincreasing order of density.

- How much can the greedy fill in any bin?

Highest density item that fits:

Density ≈ 2, wt $= 1$, size $= \frac{1}{2}+\epsilon$.

Next: Density $\approx 3/2$, wt $= 1/2$, size $= \frac{1}{3}+\epsilon$

Total size $= (\frac{1}{2}+\epsilon)+(\frac{1}{3}+\epsilon) = 5/6 + 2\epsilon$

Next item that fits in remaing $(1 - 5/6 - 2\epsilon)$ space:

Density $\approx 7/6$. wt $= 1/6$. size $= \frac{1}{7} + \epsilon$.

Total size $= \frac{5}{6} + 2\epsilon + \frac{1}{7} + \epsilon = \frac{41}{42} + 3\epsilon$

Next item that fits in remaing $(1 - \frac{41}{42} - 3\epsilon)$ space:

Density $\approx 43/42$. wt $= 1/42$. size $= \frac{1}{43} + \epsilon$.

Total size $= \frac{41}{42} + \frac{1}{43} + 4\epsilon \approx 0.999\ldots$

Total wt $= 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} \approx 1.6904..$

A careful analysis will give asymptotic bounds to be:

$$1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \frac{1}{42.43} + \frac{1}{42.43.(42.43+1)} \approx 1.691$$

It turns out this is the best:

E.g. if there is no item of class 1.

Density $\leq 3/2$. $\Rightarrow$ There must be one item from class 1.
Worst-case size $= \frac{1}{2} + \epsilon$

If there is one item of size $\frac{1}{2} + \epsilon$, & no item of class 2.

There can be at most one item $\frac{1}{4} + \epsilon$ from class 3.

Density of other classes $\leq 5/4$.

Weight $= 1 + \frac{1}{3} + \frac{5}{4} \cdot \frac{1}{4} \approx 1.64 \rightarrow$ There must be one item of size $1/3 + \epsilon$

Lower bound sequence:

$\frac{1}{43} + \epsilon$ (m items), $\frac{1}{7} + \epsilon$ (m items), $\frac{1}{3} + \epsilon$ (m items), $\frac{1}{2} + \epsilon$ (m items)

Harmonic $= m(1/42 + 1/6 + 1/2 + 1) \approx 1.691\,m$, OPT $= m$.

- FF competitive ratio can be proven to be 1.7. by following weights & case analysis:

$$
\begin{aligned}
W(x) &= 6/5\, x && \text{for } x \in [0, 1/6] \\
&= 9/5\, x - 1/10 && \text{for } x \in (1/6, \tfrac{1}{3}] \\
&= 6/5\, x + 1/10 && \text{for } x \in (1/3, \tfrac{1}{2}] \\
&= 6/5\, x + 4/10 && \text{for } x \in (1/2, 1]
\end{aligned}
$$

⊙ Present Best Bounds for Bin packing:

Algorithm [can keep unbounded number of bins open]

1.57829 [ESA'18]

Hardness:

1.54278 [Algorithmica'21]

Balogh, Békési, Dósa, Epstein, Levin.

For random-order: 3/2 [Kenyon, SODA'95]

[Conjecture: Best-Fit gives 1.15 in this case].