The Bin Packing Problem

The bin packing problem is an optimization problem, in which items of different sizes must be packed into a finite number of bins or containers each of fixed given capacity, in a way that minimizes the number of bins used.

The Decision version of the problem is NP-Complete.

Example:

bin capacity = 10

item		size	
1	–	7	1st bin
2	–	3	
3	–	6	2nd bin
4	–	4	

\# of bins required = 2

We study simplified version of the Problem.

Problem:

Given n items,

item i has size $S_i \in (0,1]$

Pack items into the fewest unit capacity bins.

Example:

item	size
1	0.6
2	0.7
3	0.8
4	0.8
5	0.3

bin capacity = 1

# of bins required = 4

Naive Algorithm - Next fit (NF) algorithm

Check to see if the current item fits in the current bin. If so, then place it there. Otherwise start a new bin.

Example: 0.5, 0.7, 0.5, 0.2, 0.4, 0.2, 0.5, 0.1, 0.6

[items are arbitrarily ordered]

OPT = 4

0.5	0.2	0.1	0.6
0.5	0.7	0.5	0.2
		0.4	

Next Fit = 6

0.5	0.7	0.2	0.2	0.1	0.6
		0.5	0.4	0.5	

Clearly NF algorithm runs in polynomial time.

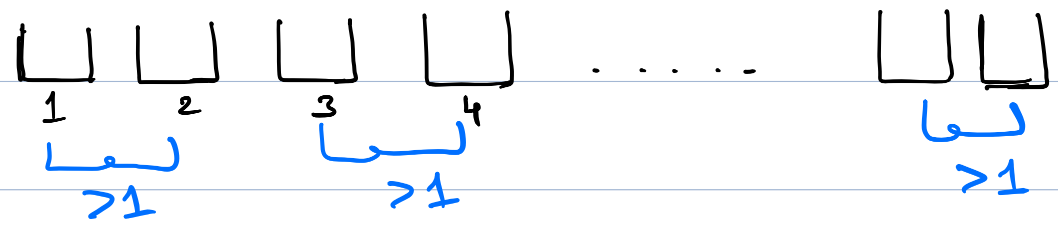
Q: How bad it is? (Approximation ratio)

Intuition: (Assume # bins is even)

Sizes of items in bin i + next item > 1

Sizes of items in bin i and bin $i+1$ > 1

Size



if Next-Fit uses k bins then

$$OPT \geq \frac{k}{2}$$

ie, $k \leq 2\,OPT$.

if # of bins is odd, then $OPT \geq \frac{k-1}{2}$

$$k \leq 2\,OPT + 1$$

Asymptotic 2-approximation.

There are several other approximation algorithms similar to NextFit. I will list few of them below.

Next-k-Fit: instead of keeping only one bin open, the algorithm keeps the last k bins open and chooses the first bin in which the item fits.

First-Fit: keeps all bins open, in the order in which they were opened. It attempts to place each new item into the first bin in which it fits.

Approximation ratio $\cong 1.7$ [Easy analysis shows it is a 2-approximation]

Best-Fit: keeps all bins open. It attempts to place each new item into the bin with maximum load in which it fits.

Approximation ratio $\cong 1.7$

FIRST FIT DECREASING (FFD)

Order the items such that $S_1 \geq S_2 \geq \dots \geq S_n$

and apply FIRST FIT.

FFD Analysis: k : # of bins used by FFD algorithm

k^* : optimal number of bins

We partition the items according to their value as follows.

$$A = \{ S_i : S_i > 2/3 \}$$

$$B = \{ S_i : \frac{1}{2} < S_i \leq \frac{2}{3} \}$$

$$C = \{ S_i : \frac{1}{3} < S_i \leq \frac{1}{2} \}$$

$$D = \{ S_i : S_i \leq 1/3 \}$$

Case 1: There is one bin b with all items from D.

In this case we know that

- b has to be the last bin
- All bins except the last bin have used more than $\frac{2}{3}$ of their capacities, otherwise items from D can be fit into them

$$\frac{2}{3}(K-1) \leq \sum_i S_i \leq OPT$$

ie, $K \leq \frac{3}{2} OPT + 1$

Case 2: If there are $t > 1$ bins with all items from D.

Above inequality still holds in this case.

$\therefore$ We can reach the same conclusion.

Case3: There is no bin with all items from D.

In this case, we remove all items of D without changing the total number of bins.

Then we have

1. No bin has more than two items
2. Any bin with one item from A can't accommodate any other item.
3. Any bin with one item from B can accommodate only another item from C.
4. Any bin with one item from C can accommodate either one item from B or one item C, but not both

As FFD Processes items by non-increasing order with respect to their weight. Therefore it Puts each item from C with the largest Possible item from B that might fit with it and that does not already share a bin with another item. Hence in this case soln of FFD and optimal solution are same.

Partition Problem:

Given a multiset S of positive integers, decide whether S can be partitioned into two sets S_1 and S_2 such that the sum of the numbers in S_1 equals the sum of the numbers in S_2.

- Partition Problem is NP-complete.

- Partition $\leq_p$ BIN-Packing

An instance $S = \{S_1, S_2, \dots S_n\}$ of Partition is YES instance if and only if the items can be packed into two bins of size $\frac{1}{2} \sum S_i$.

$\therefore$ BIN-Packing is NP-Complete.

Lemma: There is no ρ-approximation algorithm with $\rho < \frac{3}{2}$ for BIN PACKING unless $P = NP$.

Proof Same reduction as the above.

A $\rho = \left(\frac{3}{2} - \varepsilon\right)$-approximation algorithm Alg for BIN PACKING would yield a Polynomial-time algorithm for Partition Problem.

On NO-instances Alg would clearly use at least 3 bins, but on YES-instances it would use at most $\left(\frac{3}{2} - \varepsilon\right) 2 < 3$ bins.

Corollary: There is no PTAS for Bin Packing unless $P = NP$.