Approximation Algorithms - IV

Thursday, April 27, 2023 1:55 PM

Bin-Packing

Given:- Set of n items of size $a_1, a_2, \ldots, a_n$ s.t $a_i \in [0,1]$.

Goal:- Find the minimum # bins with which we can pack all items. Each bin has capacity 1.

BIN-PACKING IS NP-HARD

PARTITION $\leq_p$ BIN-PACKING

[I: set of n integers $s_1, s_2 \ldots s_n$

∃ partition S_1, S_2 s.t $\sum_{s \in S_1} s = \sum_{s \in S_2} s = \frac{1}{2} \sum_{s \in S} s$.

I': $a_i = \frac{2 s_i}{\sum_{i \in [n]} s_i} \in [0,1]$

∃ partition S_1, S_2 of S $\Longleftrightarrow$ ∃ packing into 2 bins

"$\Rightarrow$"

"$\Rightarrow$"

"$\Leftarrow$"

Question :- $\exists$ α-approximation algo for bin-packing $\alpha < \frac{3}{2}$.

Assum yes.

OPT uses 2 bins on instance I' $\Longleftrightarrow$ ALG uses 2 bins on instance I'

$\Rightarrow$ $\exists$ partition $S_1, S_2 \in S$ of instance I $\Longleftrightarrow$ ALG uses 2 bins on instance I'

Poly-nomial time algo.

HARDNESS of APPROXIMATION

Thm :- Bin-Packing does not admit α-approx. algorithm with $\alpha < \frac{3}{2}$ unless P=NP.

"amplify the gap"

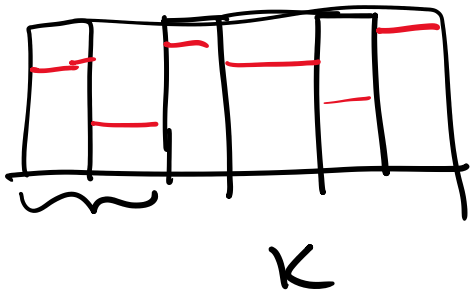
ALG (First-Fit)

while $\exists$ unallocated item i

assign i to the first bin

assign i to the first bin that i can be assigned without exceeding capacity

if no such bin exists, open a new bin & assign i to it.



K = # bins opened by greedy.

$$OPT \geq \sum_i a_i$$

$$vol(i) + vol(i+1) \geq 1. \quad \forall\, i \in [K]$$

$$\Rightarrow 2\,vol(1) + 2\,vol(K) + 2 \sum_{i \in (1,K)} vol(i) \geq K.$$

$$\Rightarrow 2 \sum_{i \in [n]} a_i \geq K.$$

bins opened by our algorithm $\leq 2 \sum_{i \in [n]} a_i \leq 2OPT$

Asymptotically Polynomial Time Approximation Scheme (APTAS)

Scheme (APTAS)

$\#\text{bins}_{\text{Alg}} \leq (1+O(\varepsilon))\,OPT + b.$

---

1) Size of every item $\geq \varepsilon$. || relax

2) # distinct item types = K.

idea :- "Brute-force over all allocation"

Polynomial

items that can be assigned to a bin $\leq 1/\varepsilon$

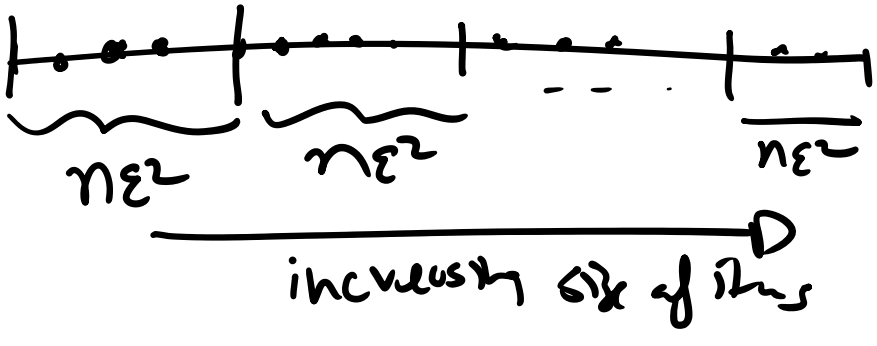
$$\#\text{ bin-types} = \binom{K+1/\varepsilon-1}{K-1} \approx O\left(K^{1/\varepsilon}\right)$$

$$\#\text{ allocations} \leq O\left(n^{K^{1/\varepsilon}}\right) \cdot \text{allocn}$$

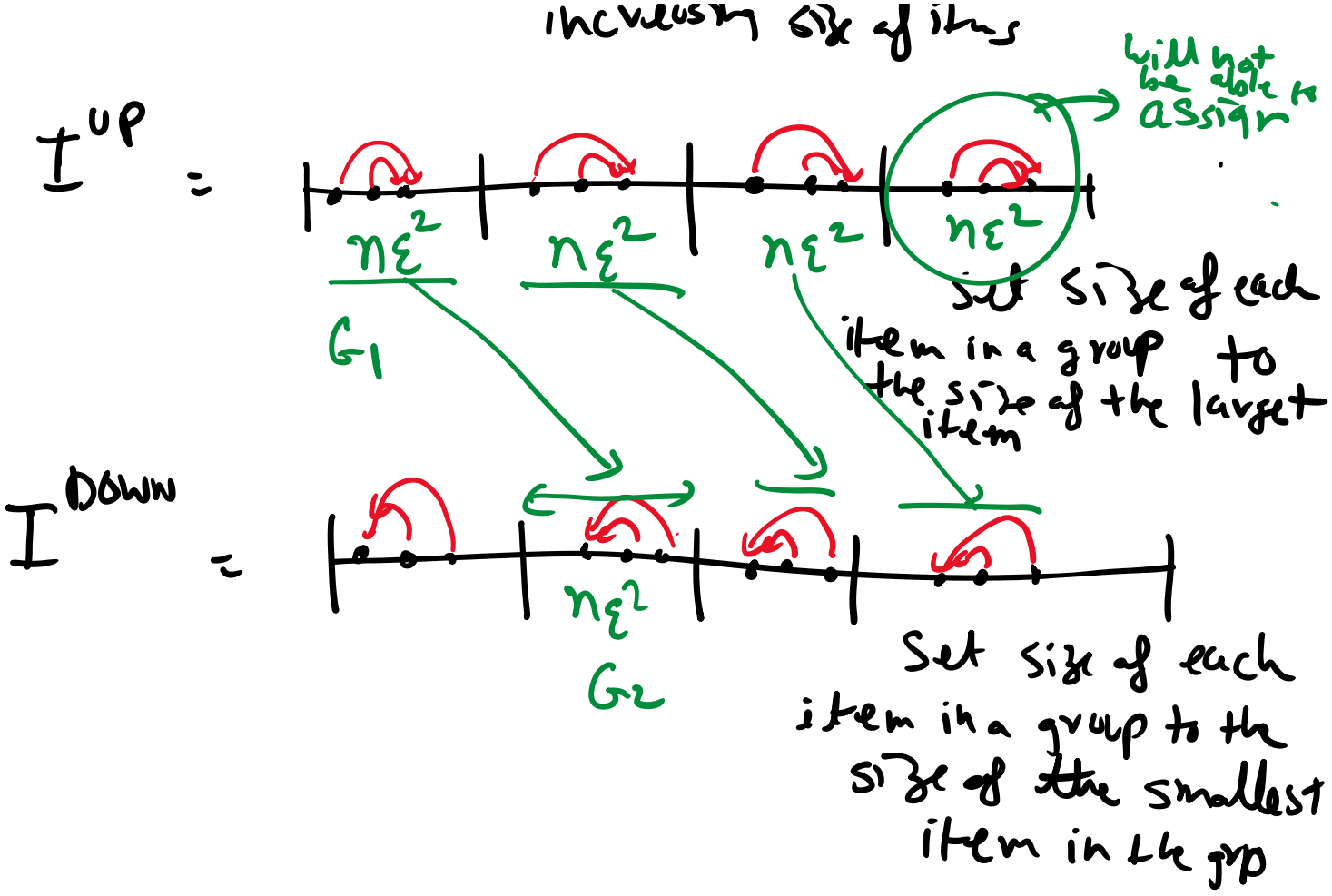
$\in$ Poly(n)

Sort items

groups $= 1/\varepsilon^2$.



will not be shown



OPT_ε =

OPT_ε^{UP}

OPT_ε^{DOWN}

$$OPT_\varepsilon^{DOWN} \le OPT_\varepsilon \le OPT_\varepsilon^{UP}.$$

#bins opened by following the solution OPT_ε^{DOWN}

$$\le OPT_\varepsilon^{DOWN} + n\varepsilon^2$$

$$= OPT_\varepsilon^{DOWN} + \varepsilon(n\varepsilon)$$

$$\le OPT_\varepsilon + \varepsilon \cdot OPT_\varepsilon$$

$\leq OPT_\varepsilon + \varepsilon \cdot OPT_\varepsilon$

$= (1+\varepsilon) OPT_\varepsilon$

ALG (for the general problem)

1) Discard S = set of items with size $\leq \varepsilon$.

2) Run alg above to find $(1+\varepsilon) \cdot OPT_\varepsilon$ many bins to allocate $[n] \setminus S$ items

3) Run First fit with items in S.

If no extra bin is opened,

$\Rightarrow$ # bins $\leq (1+\varepsilon) OPT_\varepsilon \leq (1+\varepsilon) OPT$

If extra bins were opened

bins opened $\leq (1+\varepsilon) \cdot OPT + 1$.

in the end, a total of k-bins are open,

$(k-1)$ bins are filled up to a level of $(1-\varepsilon)$

($k-1$) bins are filled up to a fact of $(1-\varepsilon)$

$$(1-\varepsilon)(k-1) \le \sum_{k-1} a_i \le OPT$$

$$\Rightarrow k \le (1+\varepsilon)OPT + 1$$

SCHEDULING :→ Given :→ set of n jobs with processing time $p_1, p_2, \ldots, p_n$

Goal :→ minimize the max-span

1) Given a threshold = t, ∃ allocation of jobs to machines with makespan $\le t$?

$$\text{Bin-packing}(I, t) \le m \ ?$$

feasible t

$$\text{min-max-span} = \min_t \{\text{Bin-packing}(I, t) \le m\}$$

$= \alpha(I, t, \varepsilon) \le m$

ALG :-

1) Discard $S = \{j \mid p_j \le t\varepsilon\}$

2) Group jobs in $[n] \setminus S$ as follows

Processing-time Rounded Proc-time

Processing-time Rounded Proc-time

$G_1 = [t\varepsilon, t\varepsilon(1+\varepsilon))$ $t\varepsilon$

$G_2 = [t\varepsilon(1+\varepsilon), t\varepsilon(1+\varepsilon)^2)$ $t\varepsilon(1+\varepsilon)$

$\vdots$

$G_i = [t\varepsilon(1+\varepsilon)^{i-1}, t\varepsilon(1+\varepsilon)^i)$ $t\varepsilon(1+\varepsilon)^{i-1}$

$\vdots$

$G_K = [t\varepsilon(1+\varepsilon)^{K-1}, t\varepsilon(1+\varepsilon)^K]$ $K \in O(\log_{1+\varepsilon} \frac{1}{\varepsilon})$

3) Solve rounded instance exactly

4) Assign jobs in S by First fit.

Bin-Packing with fixed item types t

$\text{Bins}(i_1, i_2, \ldots, i_k)$

$\sum_{j=1}^{k} i_j = n \in O\left(\binom{n+k-1}{k-1}\right) \in O(n^k)$

Initialization: To Find all $\underline{q} = (q_1, q_2, \ldots, q_k)$

s.t. $\text{Bins}(q_1, q_2, \ldots, q_k) = 1$

$|Q| \in O(n^k)$

Recurrence :-

$$O(n^k) \quad Bins(i_1, i_2, \ldots, i_k) = 1 + \min_{q \in Q} Bins(i_1 - q_1, i_2 - q_2, \ldots, i_k - q_k)$$

$$O(n^{2k})$$