Check Williamson & Shmoys

From result,

$$L^*_{max} \geqslant r(S) + p(S) - d(S)$$

$$\geqslant r(S) + p(S)$$

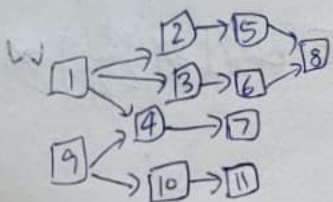
$$\Rightarrow L^*_{max} \geqslant C_j \quad - (1)$$

Also,

$$L^*_{max} \geqslant -d_j \quad - (2)$$

$$\Rightarrow 2L^*_{max} \geqslant C_j - d_j$$

- Scheduling jobs with prerequisites



- each job 1 unit time
- M machines

Dumb greedy algo – Approximation factor = 2

Proof:

- TSP

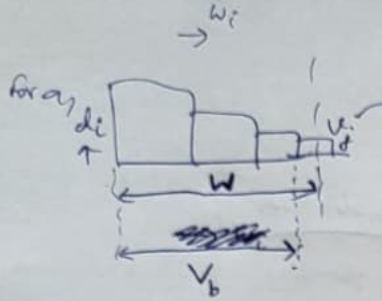
2-factor method → using MST, odd degree vertices, perfect matching,

3/2-factor

- **Knapsack** $w_1, w_2, \dots, w_n$
  $v_1, v_2, \dots, v_n$

Algo a → pick max value first

Algo b → pick max density first



then $v_j \leq V_a$ (obv)

&

So, $V_b + v_j \geq$ Max possible (if cutting is allowed) $\geq$ OPT

$\rightarrow V_b + V_a \geq OPT$

$\rightarrow \quad \text{Max}(V_a, V_b) \geq \frac{1}{2} OPT$

- **Bin Packing**

  $S = (s_1, s_2, \dots, s_n) \leftarrow$ sizes of objects $0 < s_i \leq 1$

  find Min # of bins (of size 1) needed?

Our greedy algo → keep fitting largest possible obj in bin → this algo has approx. factor $\frac{4}{3} + \frac{1}{3(OPT)}$

Proof:

Take $s_1 \geq s_2 \dots \geq s_n$

~~[illegible]~~ Let s_i be the 1st item that went into $(OPT+1)^{th}$ bin

Claim: $s_i \leq 1/3$

Assume $s_i > 1/3$

So, $s_1, s_2, \dots s_{i-1} > \frac{1}{3}$ Hence, each bin could have atmost 2 items

So, $\exists K$ s.t. $\underbrace{1\ 2 \dots K}_{\text{1 item each}}\ \underbrace{K{+}1 \dots OPT}_{\text{2 item each}} \leftarrow$ bins

$0 \leq K \leq \lceil \frac{OPT}{2} \rceil$

dabaalyuu

dablu

dabalyu

dablyu

30 time knowcheck
3 courses/sem
14 weeks/sem
1260 times/sem
8 sems in college
10080 times total
0.2 sec/syllable
3 vs 2 syll
0.2 saved/time
2016 sec saved
33.6 min saved

• Scheduling jobs with deadline on single machine

$J_i \to$ r_i (arrival), p_i (time to finish)
d_i (deadline), c_i completion time, then $l_i = c_i - d_i$ ← lateness

laugh loaf

objective : minimise $\max\{l_i\}$

Result :
Let S be a subset of jobs

$$r(S) = \min_{j \in S} r_j$$

$$p(S) = \sum_{j \in S} p_j$$

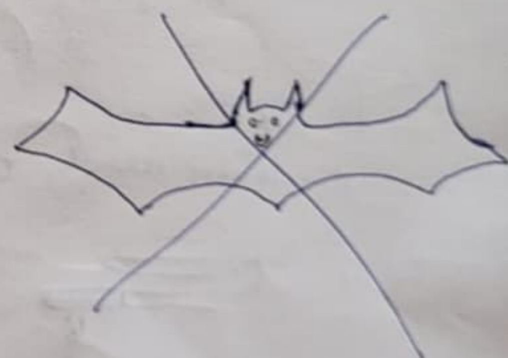
Then, $L^*_{max} \geq r(S) + p(S) - d(S)$

$$d(S) = \max_{j \in S} d_j$$

Done by the devil

~~Let~~ $L^*_{max} \to$ optimal value

Proof : let j be last job in S to be processed



Then, $d_j \leq d(S)$ — (1)

and $r(S) + p(S) \leq c_j$ — (2)

Then, $c_j - d_j \leq L^*_{max}$

$\Rightarrow r(S) + p(S) - d(S) \leq L^*_{max}$ (From (1) and (2))

$t \to \to \to \cdots \to c_j$ from $[t, c_j)$,
let S be set of jobs executed

↑ set of continously executed jobs

then, $r(S) = t$
$p(S) = c_j - t$
$= c_j - r(S) \Rightarrow c_j = p(S) + r(S)$

- **Set cover**

Input: (X, F) $\quad X = \{x_1, \ldots, x_n\}$

$$F \subseteq \mathcal{P}(X) \ni \bigcup_{C \in F} C = X$$

Output: $F' \subseteq F \ni \bigcup_{C \in F'} C = X$, where F' has minimum size.

- **Max Sat**

Given $\phi(x_1, \ldots, x_n) = C_1 \wedge C_2 \cdots \wedge C_l$

$$\text{weights} \Rightarrow \; w_1 \quad w_2 \quad \cdots \quad w_l$$

Find assignment that maximizes sum of weights of true /satisfying clauses

let $I_1 \to$ set of clauses true for $x_1 = x_2 \cdots = x_n = 1$

$I_2 \to$ " " " $= 0$

then, $W_1 = \sum_{C_i \in I_1} w_i$

$W_2 = \sum_{C_i \in I_2} w_i$

$$W_1 + W_2 \geq \sum_{\text{all } i} w_i$$

$$\Rightarrow \max(W_1, W_2) \geq \frac{1}{2}\left(\sum w_i\right) \geq \frac{1}{2} \text{OPT}$$

• Derandomization

Consider MaxSat algo,

$$E(\omega) = E[\omega \mid x_1 = 1]\, P(x_1 = 1) + E(\omega \mid x_1 = 0)\, P(x_1 = 0) \quad \xrightarrow{1/2} \quad = \frac{1}{2}\left[E[\omega \mid x_1 = 1] + E[\omega \mid x_1 = 0] \right]$$

if $E[\omega \mid x_1 = 1] \geqslant E[\omega \mid x_1 = 0]$,

then, $\leqslant \frac{1}{2}\left(2E[\omega \mid x_1 = 1]\right) = E[\omega \mid x_1 = 1]$

So, then we set $x_1 = 1$ (deterministically).

Now, to find $E[\omega \mid x_1 = 1/0]$

or

$E[\omega \mid x_1 = b_1, \dots x_i = b_i, x_i = 1]$,

$$= \sum_{j=1}^{m} w_j\, \underbrace{P(j^{th} \text{ clause is satisfied})}_{\substack{\text{will be } 1 \text{ or} \\ 1 - \left(\frac{1}{2}\right)^{l_j},\ \in \{x_{i+1}, \dots, x_n\}}}$$

similarly, at $(i+1)^{th}$ step we check if,

$$E[\omega \mid x_1 = b_1, x_2 = b_2, \dots, x_i = b_i, x_{i+1} = 1] \geqslant E[\omega \mid x_1 = b_1, \dots x_i = b_i, x_{i+1} = 0]$$

In randomized algo, to improve app. factor,

Consider maxSAT,

take $x_i = 1$ with probability $p > \frac{1}{2}$

If $C_i = (\underset{\substack{\text{(in direct} \\ \text{form)}}}{\underline{\quad a \quad}}\ \underset{\substack{\text{(in complement} \\ \text{form)}}}{\underline{\quad b \quad}})$, then $P(C_i \text{ is satisfied}) = 1 - p^b(1-p)^a$ $\quad (p > 1-p)$

$\geqslant 1 - p^{a+b}$ $\quad (a+b \geqslant 2,\ p^{a+b} \leq p^2,\ -p^{a+b} \geqslant -p^2)$

$\geqslant 1 - p^2$

(assuming $\geqslant 2$ literals)

So, best $p = \frac{\sqrt{5}-1}{2} \approx 0.61$

If single (direct) literal, $P() \geqslant p$ → handle single literal clauses seperately

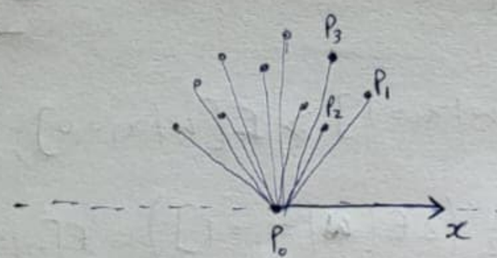
So, if we take $p' = \min(p, 1-p^2)$

Then, $E[\omega] = \sum_{i=1}^{m} w_i P(i^{th} \text{clause true}) = p' \sum w_i \geqslant p'(OPT)$

→ Computational Geometry

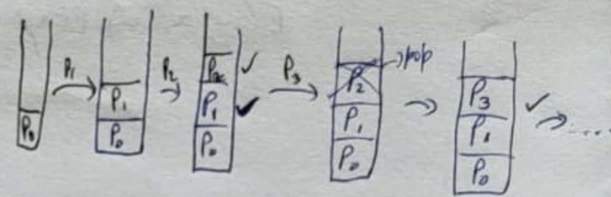
• Convex hull

Grahams scan -



Algo

Maintain stack



Add elements in order of angles (from P_0)

Pop if angle from top element is smaller

(cascading check)

Time complexity → $O(n\log n)$

Check correctness in CLRS

Jarvis' march -

↳ if given that the hull has only h points, can we do in $O(n)$ time?

Algo

Start with P_0 (lowest y-coord) (n time)

Find P_1 (lowest angle from P_0) (n time)

Find P_2 (next smallest angle from P_1) (n time)

⋮

Find P_{h-1}

→ Total ~~n~~ hn time,

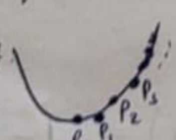
so, $O(n)^2$

• Merging 2 convex hulls (check youtube)

• Proof that lower bound on convex hull is $O(n\log n)$

↳ construct algo from hull to sorting

$x^2 = y$



so, sorting → $\{x_1, \dots, x_n\}$

$\Downarrow$

Convex hull → $P = \{(x_1, x_1^2), \dots, (x_n, x_n^2)\}$

↓ solve in $O(n\log n)$

get convex hull $P' = \{(x_1', x_1'^2) \dots (x_n', x_n'^2)\}$

↓

Sorted array $\{x_1', \dots x_n'\}$

Then, if our algo gives ALGO as sol^n,

$$ALGO \leq OPT + \left\lceil \frac{OPT-1}{3} \right\rceil$$

$$\Rightarrow \frac{ALGO}{OPT} \leq 1 + \frac{1}{OPT}\left\lceil \frac{OPT-1}{3} \right\rceil \leq 1 + \frac{(OPT+1)}{3OPT}$$

$$\leq \frac{4}{3} + \frac{1}{3OPT}$$ ← approx. factor

- <u>Scheduling Problem</u>

n jobs $J_1 \ldots J_n$, m identical machines

Times $p_1 \ldots p_n$

<u>Scheme for $(1+\epsilon)$ factor</u>

<u>Long job</u>

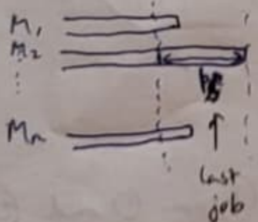
$$p_e > \frac{1}{Km} \sum_{j=1}^{n} p_j$$ others short job

$\Rightarrow$ # of long jobs $\leq km$

So, for long jobs,

do exhaustive search, $O(m^{Km})$, then for short jobs,

Then let



$$ALGO \leq p_{l} + \sum_{j=1}^{n} p_j/m$$

$$\Rightarrow ALGO \leq \sum_{j=1}^{n} \frac{p_j}{mK} + \sum_{j=1}^{n} \frac{p_j}{m}$$

$$\leq \left(1+\frac{1}{K}\right) \sum_{j=1}^{n} p_j/m$$

$$\leq \left(1+\frac{1}{K}\right) OPT = \underbrace{(1+\epsilon)}_{\text{factor}} OPT$$

$\left[\epsilon = \frac{1}{K}\right]$ with time complexity

$O(m^{1/\epsilon})$

$(ANS - p_i) \leq OPT$

$ANS < 2 * OPT$

jirro

~~jirro~~

Williamson

~~[illegible]~~

Shmoys

$\bar{x}_1 \quad x_2 \quad x_7$

→ Randomized Algorithms

- Quicksort (see in CLRS)

- MaxSAT

↳ $\phi(x_1, \ldots, x_n) = C_1 \wedge C_2 \cdots \wedge C_m$

$\qquad\qquad\qquad\qquad w_1 \quad w_2 \quad \ldots \quad w_m$

Algo ↳ set $x_i = 1$ with probability $1/2$

Then, $E[\underset{\substack{\uparrow \\ \text{total weight of algo}}}{W}] = E[\sum w_i \underset{\substack{\uparrow \\ \text{indicator if } i^{th} \text{ class is true}}}{I_i}] = \sum_{i=1}^{m} E[I_i]\, w_i$

$= \sum_{i=1}^{m} w_i\, P(C_i \text{ is true})$

$= \sum_{i=1}^{m} w_i \left(1 - \frac{1}{2^{l_i}}\right) \geq \sum_{i=1}^{m} w_i \left(\frac{1}{2}\right) \geq \frac{1}{2} W_{opt} \geq \frac{1}{2} OPT$ //

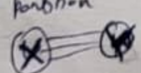
If 3-SAT, then factor will be $1 - \frac{1}{2^3} = \frac{7}{8}$ //

- Max Cut

↳ $G = (V, E)$

$\quad w : E \to \mathbb{R}^+$

partition



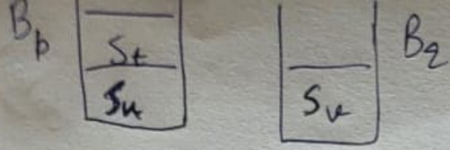
Algo ↳ For $v \in V$, let $v \in X$ with probability $1/2$

$\qquad\qquad\qquad\quad v \in Y$ " " "

So, $E[\underset{\substack{\uparrow \\ \text{of algo}}}{W}] = \sum_{(i,j) \in E} w_{ij}\, P((i,j) \text{ in cut})$

$= \sum w_{ij} \left[\underbrace{P(i \in X, j \in Y)}_{1/4} + \underbrace{P(i \in Y, j \in X)}_{1/4}\right]$

$= \frac{1}{2} \sum w_{ij} \geq \frac{1}{2} OPT$ //



where $u, v, t < i$, $u < v$

So, $S_u \geqslant S_v$

$S_t \geqslant S_i$

$S_{i-2} + S_{i-1} + S_i > 1$

$p < q < OPT$

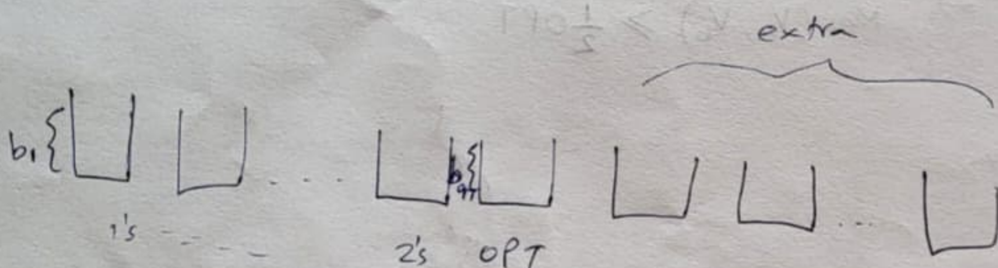
Also, $S_v + S_i > 1$ (since i went to OPT+1 bin)

Also, $S_t + S_u \leq 1$ (fitting in bin)

But $1 \geqslant S_t + S_u \geqslant S_i + S_v > 1 \rightarrow$ contradiction

Hence, $S_i \leq 1/3$

Now, Claim, i.e. # of items in extra bins is atmost OPT − 1



Proof:

$$\sum_{i=1}^{\hat{}} S_i \leq OPT \quad - (1)$$

Let items in extra bins be $t_1, t_2, \ldots t_{OPT}$ (suppose there are OPT items)

So, $b_i + t_i > 1$, $1 \leq i \leq OPT$ − (2) (from the algo)

$$\sum_{i=1}^{\hat{}} S_i \underset{(=)}{\geqslant} \sum_{j=1}^{OPT} b_j + \sum_{j=1}^{OPT} t_j = \sum_{j=1}^{OPT} b_j + t_j > OPT$$

So, contradiction.

Hence atmost OPT − 1 items

text → t = " ... " → length N How many times does p occur in t? (and which pos^n)

pattern → p = " ... " → length M

Brute force – $O(MN)$

KMP – $O(N)$

text $\;1\;2 \;\ldots\; i{-}1\;\; i\;\; i{+}1$

pat $\;1 \;\ldots\; i$

Method:

Let $lps[i]$ = length of longest prefix of pat[0...i] that is also suffix of ↗

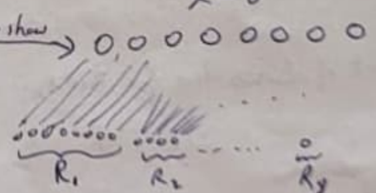
- Approximation Algorithms
  - correctness
  - time comp.
  - approximation factor

Heuristic algorithms

- Vertex-cover

  └ Initial method – (highest degree first) $\xrightarrow{\text{counterexample to show}}$

$l = 8$



Proud of youuu :D (that is me) ☆☆

R_i group has $\lfloor \frac{l}{i} \rfloor$ nodes, each with degree i

Here opt $= l \,(=8)$, but algo $= \sum_{i=1}^{l} \lfloor \frac{l}{i} \rfloor \leq l \log l$

So $\frac{\text{algo}}{\text{opt}} = \log l \to \infty$ as $l \to \infty$

– solution (with app. factor = 2)

Proof: (using maximal matching argument)

- Weighted vertex cover

  └ frame as ILP → convert to LP → solve → convert back to sol^n of weighted VC with approx factor = 2.