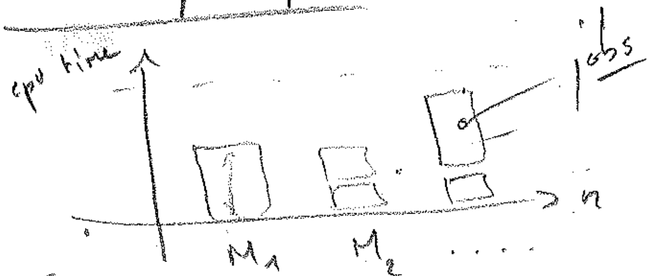
Algorithm Design

o Makespan pb



online greedy

<u>sol^o_1</u> place job on least loaded machine
<u>approx</u> $= 2 - \frac{1}{m}$

<u>sol^o_2</u> order jobs by load value
<u>approx</u> $= \frac{4}{3} - \frac{1}{3m}$

this is the best pure greedy

$\boxed{\text{PTAS}}$ Polynomial Time Approximation Scheme

brute-force greedy

online greedy is best for $m = 2$, $m = 3$

→ use brute force to sort input to optimal ordering before processing with LPT

Other LPTs → mach. w/ ≠ speeds (uniform)
→ restricted model : job can only be run on specific mach.
→ unrelated mach. model. : jobs of vectors of processing time on ≠ mach.

o Knapsack pb

<u>greedy</u> → $\frac{\$}{size}$ ration, ↗\$ ∴ not optimal

partial enumeration greedy

→ start with HC I ⟶ sorted items by $\frac{\$}{size}$ ratio
↳ approx $= 2$

⇒ FPTAS → approx $= (1+\varepsilon)$

$$V \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = t \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \rightarrow Null\,A = span\left(\begin{smallmatrix}1\\0\\1\end{smallmatrix}\right)$$

$$\begin{array}{c} x_1 \; x_2 \; x_3 \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} 0 \\ 0 \\ 0 \end{matrix}, \quad x_3 = t$$

$$\rightarrow \begin{cases} x_1 = -t \\ x_2 = 0 \\ x_3 = t \end{cases} \rightarrow t[\;]$$

proof of Sahni's PTAS

$\left[|OPT| \le k \Rightarrow PGreedy_k \text{ is optimal} \right] \Rightarrow |OPT| > k$

H = most profitable k items in OPT

$L = OPT - H$ $(\neq \emptyset)$ $\rightarrow H = \overbrace{\{J_0 \dots J_k\}}^{\text{most profitable items}}$

$L = J_{k+1} \dots J_R$

→ if all of L is packed by PGreedy, then PGreedy is optimal.

So let J_m be first item that doesn't fit in knapsack

→ with value V_m

$k+1 \le m \le R$

→ let G be the $m-k-1$ items in $\{J_{k+1} \dots J_m\}$

$V(OPT) - V(H+G) \le V(J_m) = V_m$

claim: ∃ at least k+1 items

all items in L not packed by PGreedy → have value $\le V_m$

$\forall J_j, j > m$, profit density less than J_m.

⇒ the value of ~~the~~ any item in V that can be placed in the knapsack is less than or equal to V_m.

$\Rightarrow (k+1) V(OPT) \le V(H+G) + \frac{V(OPT)}{k+1}$

$\Rightarrow V(OPT) \le \frac{k+1}{k} V(H+G)$

$\le \left(k + \frac{1}{k}\right) V(PGreedy)$

because $V_m \le \frac{V(OPT)}{k+1}$

---

cf Church-Turing th. for Greedy Algs.

miopic algs: do not look into the future.

greedy sorting: doesn't make sense, greedy is metasorting

Priority Stack Alg

Chordal graph: ∃ ording such that for any vertex v_i; ~~its Nbh~~ is ~~a clique~~ $nbhd(v_i) \cap \{v_{i+1} \dots v_n\}$ is a clique (completely connected)

→ order intervals by finishing time PEO Perfect Elimination Ordering

side note: WISP → Greedy$_\alpha$

$v(ALG) \doteq \frac{v(T)}{1-\alpha}$, $v(ALG_i) \doteq \frac{v(T_i)}{1-\alpha}$ (by induc.), $v(OPT) \leq \frac{v(T)}{\alpha}$

(k+1)-claw free graph: $\forall v \in V$, the induced subgraph of its Nbhd(v_i) has at most k indep. vertices.
⇒ it doesn't have a (k+1) claw $\}k$

claw free graph: = 3-claw free graph

→ Generalizing on (k+1)-claw free and chordal graphs:
inductively k-indep. if ∃ k-PEO of $v_1 \dots v_n$, $nbhd(v_i) \cap \{v_{i+1} \dots v_n\}$ has at most k vertices.

"Some good ppl said that ∀ polytime pb, ∃ polytime DP pb"
↳ Borodin: yes... any block can be computed by other blocks in polytime, but how?

Missed DP...

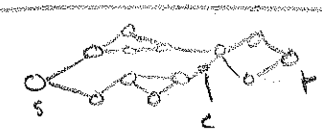
→ Local Search
1. Jump local search
2. Push local search

side-note: Linear Programming can be solved by simplex (local search of LP polytope)

Flow Networks

→ $\boxed{\text{DP cannot formalize polytime} \to \text{unweighted bipartite matching}}$ cannot be solved by DP in poly time, but flow can

paradigms
→ Greedy
→ Local Search
→ LP/IP
→ DP
→ primal/dual prog.



. $f(u,v) = -f(v,u)$
. $\forall u$, $|in flow(u)| = |out flow(u)|$

→ Max flow / Min cut theorem
⇒ any local optimum (found by loc. search) is an absolute optim.

→ Since complexity cannot tell us what ordering methods are possible in polynomial time, we only look at all interesting ordering

<u>Fixed priority alg</u> : decide total ordering π over J at first and <u>fix</u> it.

∫ nonfixed, ordering may be redefined.
↳ <u>adaptive priority alg</u>

• For weighted interval selection (packing pb) with weighted values (resp. for proportional weights $v_j = |f_j - s_j|$), no priority algorithm can achieve cst approx (resp. better than a 3-approx.)

<u>Ex</u> knapsack :

	size	profit	
L (large jobs)	1	1	
M (med.)	$1/n$	$1/\sqrt{n}$	→ $n^{1/4}$
S (small)	$1/n^2$	$1/n$	→ $\frac{n^{3/4}}{n} - \frac{1}{n^{1/4}}$

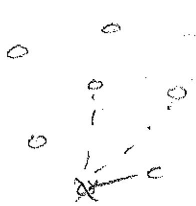
<u>Extension of priority model</u>

→ <u>priority with revocable acceptances</u>

<u>ex</u>: • Job Interval Selection Problem is NP-hard
• Weighted JISP has a 2-approx → <u>no PTAS possible</u>

<u>Greedy</u>$_\alpha$ → not greedy! → conservatively greedy.

<u>Grahams convex hull alg</u>



choose pt in convex hull (eg. leftmost) p_1
Sort remaining pts by increasing angle (p_1, p_i) : $p_2\ p_3$
stack pts
while top 3 pts make a 'right' turn, remove a pt

Catch up - LP!

Any LP can be put in standard forms:

- $\boxed{\text{primal:}}$ wlog $\begin{cases} \min\ c \cdot x \\ \text{subj to } A \cdot x \geq b \\ x \geq 0 \end{cases}$
- $\boxed{\text{dual:}}$ wlog $\begin{cases} \max\ b \cdot y \\ \text{subj to } A \cdot y \leq c \\ y \geq 0 \end{cases}$

if min then max wlog

→ note: dual(dual) = primal

Thm (strong duality) if x^* is primal optimal ∧ y^* is optimal dual, then: $P(x^*) = D(y^*)$

values of objective f°

Thm (weak duality) if x is primal sol° ∧ y is dual sol°, then $D(y) \leq P(x)$

→ f-frequency set cover pb can be solved ~~heuristically~~ w/ approx. opt (without LP) by a primal-dual algorithm (derived from theorem)

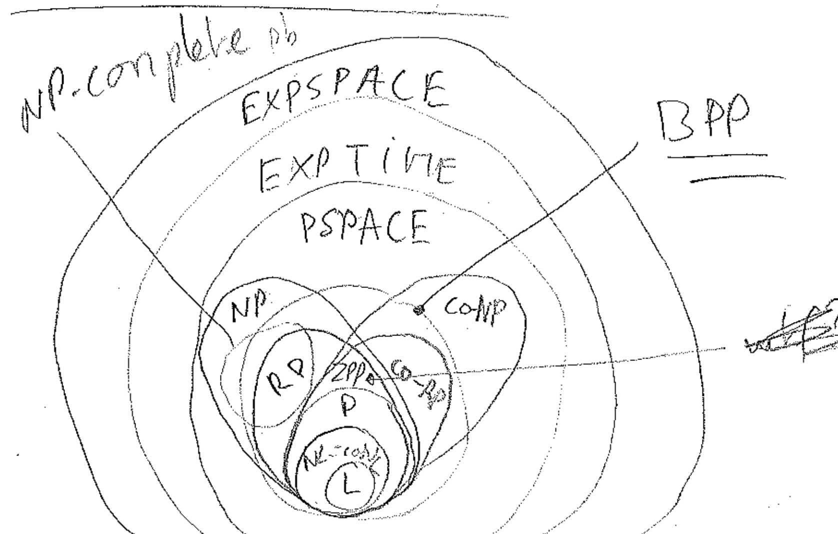
⇒ PRIMAL-DUAL ALGS are $\boxed{\text{combinatorial}}$, they help circumvent use of LP.

Note from Borodin: In heuristics we don't prove stuff about optimal approx etc. whether we do for approx algs.

Feedback vertex pb
NP-complete

FVS (feed. vert. set) is a set ⊆ G(V,E) such that if removed, the graph becomes acyclic

RANDOMIZATION



Quantum computa°

Borodin: "Cook's student said classical theoretical CS ppl live in beautiful mansion w/ founda° infested by termites"
(can't see them but might destroy it)

BPP = P ?

Borodin: "would bet that it's not true, but not too much"

Randomized rounding for MAX-SAT

Formulate weighted MAXSAT as $\{0,1\}$ IP $\Big\}$ → relax vars to $[0,1]$

$\begin{cases} \text{Maximize } \sum w_j z_j \\ \text{subject to } \sum \sim \quad \sum \sim \geq z_j \end{cases}$

Quadratic prog for MAX-2SAT

→ introduce y_0 (homogenizing var), $val(x_i) = (1+y_iy_0)/2$
$val(\bar{x}_i) = (1-y_iy_0)/2$

→ strict quadratic

→ then relax ~~vars~~ to vector program $\Big\{ y_i \to$ unit $v_i \in \mathbb{R}^{n+1}$

→ then randomize w/ hyperplanes $\Big\}$ $* \to$ dot prod.

Random walk for 2SAT

~

→ walk sat : heuristics from random walks on k SAT

→ why is this local search? isn't neighborhood property ~

| involution is a function that is its own inverse |

$\tilde{O}$-notation : hide $\begin{cases} \text{poly } \frac{1}{\varepsilon} \leftarrow \text{errorbound} \\ \text{polylog } m, n, \frac{1}{\delta} \leftarrow \text{confidence bound} \end{cases}$

250 → 302 → 303 → 410
247 → 347 → 447 → 414

Weighted Majority Algorithm   Arora, Hazan, Kale 2005
(cf random sampling)

cf spectral methods → Laplacian of adj, get eigvals

Submodular maximization

def: $f: 2^U \to \mathbb{R}$, $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ when $S, T \subseteq U$

non-mono ⇒ 1/2 approx
mono ⇒ $1-1/e$ approx

$(\Leftrightarrow) \underbrace{f(S \cup \{x\})}_{-f(S)} \geq \underbrace{f(T \cup \{x\})}_{-f(T)}$ when $S \subseteq T \subseteq U$

Matroid generalizes independence (linear)

when $|X| < |Y|$ but $w(X) = w(Y)$ to get X (better) we $w \leftarrow \Delta$

| assuming RP ≠ NP or using oracles are = ysbounds |

| note max-cut $S \sim \bar{S}$ f: submodular |

proof for USM's deterministic 1/3 approx (4)

$$\begin{cases} a_i = f(X_{i-1} \cup \{u_i\}) - f(X_{i-1}) \\ b_i = f(Y_{i-1} \cup \{u_i\}) - f(Y_{i-1}) \end{cases}$$

(1) $\overline{f}$ is submodular

$\Rightarrow (X_{i-1} \cup \{u_i\}) \cup (Y_{i-1} \setminus u_i\} = Y_{i-1} \wedge (X_{i-1} \cup \{u_i\}) \cap (Y_{i-1} \setminus u_i) = X_{i-1}$

$$\Rightarrow \begin{cases} (X_{i-1} \cup \{u_i\}) \cup (Y_{i-1} \setminus u_i) = X_{i-1} \cup Y_{i-1} \\ (X_{i-1} \cup \{u_i\}) \cap (Y_{i-1} \setminus u_i) = X_{i-1} \cap Y_{i-1} \end{cases}$$

(2) $a_i + b_i = [f(X_{i-1} \cup \{u_i\}) - f(X_{i-1})] + [f(Y_{i-1} \setminus \{u_i\}) - f(Y_{i-1})]$

$= [f(X_{i-1} \cup \{u_i\}) + f(Y_{i-1} \setminus u_i)] - [F(X_{i-1}) + f(Y_{i-1})]$

≥ 0 (by submodularity)

(3) $OPT_i \overset{def}{=} (OPT \cup X_i) \cap Y_i$ coincides for X_i and Y_i on 1^{st} element of $U: \{u_1 \ldots u_i\}$

$OPT_0 = OPT \; ; \; OPT_n = X_n = Y_n$

(4) $\sum [f(OPT_{i-1}) - f(OPT_i)] \leq \sum [f(X_i) - f(X_{i-1}) + f(Y_i) + f(Y_{i-1})]$

(5) $f(OPT_0) - f(X_n) = f(OPT_0) - f(OPT_n) \leq f(X_n) - f(X_0) + f(Y_n) - f(Y_0)$

$\leq f(X_n) + f(Y_n)$

$\Rightarrow f(OPT) \leq 3 f(X_n) = 3 f(?)$

$f(OPT_{i-1}) - f(OPT_i) \leq [f(X_i) - f(X_{i-1})] + [f(Y_i) - f(Y_{i-1})]$

- case 1: $a_i \geq b_i$ (case 2: $b_i > a_i$)

$0 \leq f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\}) \leq f(X_i) - f(X_{i-1}) \leq a_i$

because $(a_i \geq b_i \wedge a_i + b_i \geq 0 \Rightarrow a_i \geq 0)$ by submod

and because : subcases: $u_i \in OPT$ ✓

$u_i \notin OPT \Rightarrow u_i \in OPT_{i-1}$ ✓

$\Rightarrow f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\}) \leq f(Y_{i-1} \setminus u_i) - f(Y_i) = b_i \leq a_i$

$u_i \in Y_{i-1} = Y_i$?

Matroid $\mathcal{M} = (E, I)$, — family of IS of E

$$\begin{cases} \emptyset \in I \ (\Leftrightarrow I \neq \emptyset) \\ \forall A' \subset A \subset E,\ A \in I \Rightarrow A' \in I & \sim \text{hereditary property} \\ A, B \in I,\ |A| > |B| \Rightarrow \exists\, a \in A,\ B \cup \{a\} \text{ is IS} & \sim \text{exchange property} \end{cases}$$

→ Lance Fortnow Moser

→ kolmogorov complexity

"as long as r is a truly random string this holds"

information theory applied to local search