Algorithm Design

- Makespan pb

*Online greedy*

- 1. Place job on least loaded machine
  \[
  \text{approx} = 2 - \frac{1}{m}
  \]
- 2. Order jobs by load value (this is the best pure greedy)
  \[
  \text{approx} = \frac{4}{3} - \frac{1}{3m}
  \]

PTAS: Polynomial Time Approximation Scheme

- Break down into online greedy, is best for \( m = 2 \), \( m = 3 \)
- Use brute force to sort input to optimal ordering before processing

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 each

- Knapsack pb

*Greedy*

- \( \frac{1}{2} \) ratio, not optimal
- Partial enumeration greedy

- Start with \( k \leq 1 \) - sort items by \( \frac{1}{2} \) ratio

\[
V(S) = \max \left\{ \sum_{i=1}^{n} w_i \cdot \frac{x_i}{\text{value}} \right\}
\]

\[
\text{OPT} = \sum_{i=1}^{n} \frac{w_i}{\text{value}} \left( x_i \right)
\]

\[
V(S) \leq \left( 1 + \epsilon \right) \sum_{i=1}^{n} \frac{w_i}{\text{value}} \left( x_i \right)
\]
proof of Sahni's PTAS

\[ \{ \text{OPT} \leq k \Rightarrow \text{PGreedy}_k \text{ is optimal} \} \Rightarrow |\text{OPT}| > k \]

- \( H = \) most profitable \( k \) items in \( \text{OPT} \)
  \( \text{most profitable items} \)

- \( L = \text{OPT} - H \) \( (\neq \emptyset) \)

- \( L = \{ J_{k+1}, \ldots, J_{m} \} \)

\( \Rightarrow H = \{ J_1, \ldots, J_k \} \)

- \( \Rightarrow \) if all of \( L \) is packed by PGreedy, then PGreedy is optimal.
- So let \( J_m \) be the first item that doesn't fit in knapsack
  \( k+1 \leq m \leq r \)
  to fill value \( V_m \)

- \( \Rightarrow \) let \( G \) be the remaining items in \( \{ J_{k+1}, \ldots, J_m \} \)

\[ V(\text{OPT}) - V(H + G) \leq V(J_m) = V_m \]

**claim:** \( \exists \) at least \( k+1 \) items
- all items in \( L \) not packed by PGreedy have value \( \leq V_m \)
- \( \forall J_j, j > m \), profit density less than \( V_m \)

\( \Rightarrow \) the value of any item \( i \) that can be placed in the knapsack is less than or equal to \( V_m \)

\[ (k+1) V(\text{OPT}) \leq V(H + G) + \frac{V(\text{OPT})}{k+1} \]

\[ \Rightarrow V(\text{OPT}) \leq \frac{k+1}{k} V(H + G) \]

because \( V_m \leq \frac{V(\text{OPT})}{k+1} \)

---

Church-Turing thesis for Greedy Alg.

Myopic alg. - do not look into the future

Greedy sorting - doesn't make sense, greedy is meta-sorting
Priority Stack Alg

Chordal graph:

- Vertex ordering such that for any vertex \( v \), its neighborhood \( \text{nbhd}(v_i) \cap \{v_1, \ldots, v_n\} \) is a clique (completely ordered)

- Order intervals by finishing a Perfect Elimination Ordering

\[
\text{OPT} = \frac{v(T)}{1-\alpha}, \quad \text{OPT} < \frac{v(T)}{1-\alpha} \quad \text{(by induction)}
\]

\( (k+1) \)-claw free graph:

- For any vertex \( v \), the induced subgraph of its neighborhood \( \text{nbhd}(v_i) \) has at most \( k \) independent vertices.
- Thus, it doesn't have a \((k+1)\) claw.

Claw free graph:

- \( k \)-claw free graph

- Generalizing on \((k+1)\)-claw free and chordal graphs:

- Inductively \( k \)-indep. if \( 3 \) \( k \)-PEO of \( v_1, \ldots, v_n \), \( \text{nbhd}(v_i) \cap \{v_1, \ldots, v_n\} \) has at most \( k \) vertices.

- Some good ppt said that V polyline pb, 3 polyline DP pb u

- Boaz: yes...

Missed DP...

- Local Search
  - Jump local search
  - Push local search

Flow networks

- f(u, v) = f(v, u) (by DP in polytime)

- Max flow / Min cut theorem

- Any local optimum (found by loc. search) is an absolute optimum.
Since complexity cannot tell us what ordering methods are possible in polynomial time, we only look at all interesting ordering.

**Fixed priority alg:** decide total ordering at once at first and fix it.

If new fixed, ordering may be redefined.

**LS adaptive priority alg**

For weighted interval selection (packing pb) with weighted values (resp. for proportion weights $v_j = 1/j$), no priority algorithm can achieve an $O(1)$ approximation (resp. better than a $1.3$-approx.)

*Example: knapsack*

<table>
<thead>
<tr>
<th>Job</th>
<th>Size</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>L (large)</td>
<td>1/2m</td>
<td>$m/2n$</td>
</tr>
<tr>
<td>M (med)</td>
<td>1/m</td>
<td>$1/n$</td>
</tr>
<tr>
<td>S (small)</td>
<td>1/m2</td>
<td>$1/m - 1/n$</td>
</tr>
</tbody>
</table>

Extension of priority model

- Priority with revocable acceptances

*Example: Job Interval Selection Problem* is NP-hard

Weighted JISP has a $2$-approx. no PTAS possible

Greedy $x$ is not greedy! is conservatively greedy.

**Graham's convex hull alg**

- Choose $p_1$ in convex hull (the leftmost) $p_i$
- Sort remaining $p_i$s by increasing angle ($\text{perp}p_i$): $p_2, p_3$
- Stack $p_i$s
- While top $3$ $p_i$s make a 'right' turn, remove a $p_i$
Catch up - LP!

Any LP can be put in standard forms:

- primal: \[ \text{wlog} \begin{cases} \min & c^T x \\ \text{subj to} & A x \leq b \\ x \geq 0 \end{cases} \]

- dual: \[ \text{wlog} \begin{cases} \max & b^T y \\ \text{subj to} & A^T y \leq c \\ y \geq 0 \end{cases} \]

\[ \iff \begin{cases} \text{min} \\ \text{max} \end{cases} \begin{cases} c^T x \\ b^T y \end{cases} \begin{cases} A x \leq b \\ A^T y \leq c \end{cases} \begin{cases} x \geq 0 \\ y \geq 0 \end{cases} \]

\[ \text{then max wlog} \]

\[ \Rightarrow \text{note: deal (dual) = primal} \]

Then, if \( x^* \) is primal optimal, \( y^* \) is optimal dual, then: \( P(x^*) = D(y^*) \)

\[ \text{values of objective for} \]

\[ \text{w/ approx opt} \]

Then, if \( x^* \) is primal sol, \( y^* \) is dual sol, then \( D(y) \leq P(x) \)

\[ \begin{cases} \text{f=frequency set cover pb} \end{cases} \]

\[ \begin{cases} \text{can be solved heuristically} \end{cases} \]

\[ \begin{cases} \text{without} \ LP \end{cases} \]

\[ \begin{cases} \text{by a primal-dual algorithm theorem} \end{cases} \]

\[ \begin{cases} \text{primal-dual algos are combinatorial} \end{cases} \]

\[ \begin{cases} \text{they help} \end{cases} \]

\[ \begin{cases} \text{circuit use of LP} \end{cases} \]

Feedback vertex pb:

\[ \begin{cases} \text{NP-complete} \end{cases} \]

Feedback vertex set (FVS) is a set \( S \subseteq V(G(V,E)) \) such that

\[ \begin{cases} \text{if removed, the graph becomes acyclic} \end{cases} \]

\[ \begin{cases} \text{NP-complete} \end{cases} \]

RANDOMIZATION

\[ \begin{cases} \text{BPP} \end{cases} \]

\[ \begin{cases} \text{BPP = P?} \end{cases} \]

Quantum computers

Borodin: "Would bet that its not too much."

Cook's student said classical theoretical that its not too much too true, but not too much "founded" infected by broken

In cannot see then can't right destroy teh
Randomized rounding for MAX-SAT

Formulate weighted MAX-SAT as \( \{0,1\}^P \)
\[
\begin{align*}
\text{Maximize} & \quad \sum w_i z_i \\
\text{subject to} & \quad \sum \geq 2j
\end{align*}
\]

Quadratic prog for MAX-2SAT

- Introduce \( y_0 \) (homogenizing var), \( \text{Val}(x_i) = (1+y_i y_0)/2 \)
- \( \text{strich quadratic} \)
- Then relax \( y_0 \) to \( \mathbb{R} \)
- Then randomize \( y_0 \) via hyperplane

Random walk for 2SAT

- To walk sat: heuristics from random walks on KSAT
- Why is this local search? Isn't it non-homogeneous?

Involution is a function that is its own inverse

\( \mathcal{W} \): widely poly \( \in \) crossovers

Weighted Majority Algorithm (Azar, Hazan, Kale 2005)

of spectral methods \( \rightarrow \) Laplacian of adj., get signals

Submodular maximize

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

\( \text{when } |X| < |Y| \text{ but } w(X) = w(Y) \text{ to get } x \text{ (better)} \)

Matroid generalizes independence (linear)

Assuming \( RR \neq NP \)

or using cuts, Minkowski sums

Matroid cuts are \( \leq \) Yesorno:\( \sum S \leq \sum S \neq \sum S \)}
proof for USM's deterministic V3 approximation

\[ a_i = f(x_{i-1} \cup \{u_i\}) - f(x_{i-1}) \]
\[ b_i = f(y_{i-1} \cup \{u_i\}) - f(y_{i-1}) \]

1. \( F \) is submodular

\[ (x_{i-1} \cup \{u_i\}) \cup (y_{i-1} \cup \{u_i\}) = x_{i-1} \cup (y_{i-1} \cup \{u_i\}) = y_{i-1} \]
\[ (x_{i-1} \cup \{u_i\}) \cap (y_{i-1} \cup \{u_i\}) = x_{i-1} \cap y_{i-1} \]

2. \( a_i + b_i = \left[ f(x_{i-1} \cup \{u_i\}) - f(x_{i-1}) \right] + \left[ f(y_{i-1} \cup \{u_i\}) - f(y_{i-1}) \right] = \left[ f(x_{i-1} \cup \{u_i\}) + f(y_{i-1} \cup \{u_i\}) \right] = \left[ f(x_{i-1}) + f(y_{i-1}) \right] \geq 0 \) (by submodularity)

3. \( \text{OPT}_i = (\text{OPT} \cup x_i) \cap y_i \) coincides for \( x_i \) and \( y_i \) on 1st element of \( U_i \) for \( U_i \neq \emptyset \)
\[ \text{OPT}_d = \text{OPT} \quad \text{OPT}_m = x_m = y_m \]

4. \[ \sum \left[ f(\text{OPT}_{i-1}) - f(\text{OPT}_i) \right] \leq \sum \left[ f(x_i) - f(x_{i-1}) + f(y_i) + f(y_{i-1}) \right] \]

5. \[ f(\text{OPT}_0) - f(x_m) = f(\text{OPT}_0) - f(\text{OPT}_m) \leq f(x_m) - f(x_i) + f(Y_m) - f(Y_i) \leq f(x_m) + f(Y_m) \]

\[ \Rightarrow f(\text{OPT}) \leq 3 f(x_m) = 3 f(\emptyset) \]

\[ f(\text{OPT}_{i-1}) - f(\text{OPT}_i) \leq \left[ f(x_i) - f(x_{i-1}) \right] + \left[ f(y_i) - f(y_{i-1}) \right] \]

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

\[ 0 \leq f(\text{OPT}_{i-1}) - f(\text{OPT}_m \cup \{u_i\}) \leq f(x_i) - f(x_{i-1}) \leq a_i \]

because \( (a_i \geq b_i \land a_i \cdot b_i \geq 0 \Rightarrow a_i \geq 0) \)

and because: subcases: \( u_i \not\in \text{OPT} \)

\[ u_i \in \text{OPT} \Rightarrow u_i \in \text{OPT}_{i-1} \]

\[ \Rightarrow f(\text{OPT}_{i-1}) - f(\text{OPT}_m \cup \{u_i\}) \leq f(Y_{i-1} \cup u_i) - f(Y_i) = b_i \leq a_i \]

\[ \sim u_i \in Y_{i-1} = Y_i \sim \]