α- Approximation Algorithms - II

$$\max\left(\frac{ALG(I)}{OPT(I)}, \frac{OPT(I)}{ALG(I)}\right) \le \alpha$$

$\boxed{\text{Greedy}}$ DP LP Randomization

Knapsack: Given: → 1) Set S of items
2) each item i has value v_i & cost c_i
3) Threshold C

Goal: → Find $S' \subseteq S$ with $\max \sum_{i \in S'} v_i$

s.t. $\sum_{i \in S'} c_i \le C$.

$DP[V, i]$ = minimum cost achievable by selecting some set of first 'i' items that have total value V.

$S = \{\underbrace{1, 2, \ldots}_{i} \ldots n\}$

$DP[V, i] = \min\{ DP[V, i-1]$ // if subset does not

$$DP[\underline{V, i}] = \min \begin{cases} DP[V, i-1] & \text{// S subset does not contain } i^{th} \text{ item} \\ DP[V - v_i, i-1] + c_i \end{cases}$$

Assume that each $v_i \leq V$

1 to $\underline{nV}$

$\#$ Table entries $= \underline{n^2 V}$

PARTITION $\leq_p$ KNAPSACK

---

1) Pick $\varepsilon > 0$

2) Round down the values. $v_i' = \left\lceil \left(\frac{\cdot v_i}{\underbrace{\varepsilon \cdot V/n}_{= b}} \right) \right\rceil$

$v_i' = \left\lceil \frac{v_i}{b} \right\rceil$

3) Run DP for knapsack on scaled instance

$\max_i v_i' \in O(n \varepsilon^{-1})$

DP runs in time $O(n^2 \cdot n \varepsilon^{-1}) \in O(n^3/\varepsilon)$

Let S be the sol^n returned by ALG

S^* be the opt. solution

$$\sum_{i\in S} v_i \geq \sum_{i\in S} b\left\lfloor \frac{v_i}{b} \right\rfloor$$

$$\geq \sum_{i\in S} b\left(\left\lceil \frac{v_i}{b} \right\rceil - 1\right)$$

$$= b\left(\sum_{i\in S} \left\lceil \frac{v_i}{b} \right\rceil\right) - b|S|$$

$$\geq b\sum_{i\in S^*} \left\lceil \frac{v_i}{b} \right\rceil - b|S|$$

$$\geq \sum_{i\in S^*} v_i - bn$$

$$= \sum_{i\in S^*} v_i - \cancel{b}\cdot\frac{\varepsilon V}{\cancel{n}}$$

$$= \sum_{i\in S^*} v_i - \varepsilon V$$

$$\geq OPT - \varepsilon\, OPT$$

$$= (1-\varepsilon)\, OPT$$

1) $(1-\varepsilon)^{-1} \sim (1+\varepsilon)$ approx alg that

1) $(1-\varepsilon)^{-1} \approx (1+\varepsilon)$ approx alg that runs in $O(n^3\varepsilon^{-1})$

Polynomial Time Approximation Scheme (PTAS)

LP based approximation algorithms

Vertex-cover

Given: 1) Graph $G=(V,E)$

Goal: Find $S \subseteq V$ s.t $\forall (u,v) \in E$
$u \in S$ or $v \in S$

$x_v = \begin{cases} 1 & \text{if } v \in S \\ 0 & \text{otherwise} \end{cases}$

Set cover

$\min \sum_S x_S w_S$

s.t $\sum_{S: e \in S} x_S \geq 1 \quad \forall e$

$x_S \in \{0,1\}$

SC: ILP

VC: ILP

VC: LP

$\min \sum_{v \in V} x_v$

s.t. $x_u + x_v \geq 1 \quad \forall (u,v) \in E$

$x_v \in \{0,1\} \quad \forall v \in V$

$0 \leq x_v \leq 1 \quad \forall v \in V$

1) SOLVE VC:LP & let x be the optimum solution

2) ROUND x to an integer solution y for VC:ILP

For each $v \in V$

if $x_v \geq 1/2$ & set $\underline{y_v = 1}$

$x_v < 1/2$ & set $\underline{y_v = 0}$

Thm: y is a 2-approximation alg for VC: ILP

$y_v \in \{0,1\} \ \forall v \in V$ ✓

$y_u + y_v \geq 1 \quad \forall (u,v) \in E$ ✓

We know that $x_u + x_v \geq 1 \Rightarrow x_u \geq 1/2 \Rightarrow y_u = 1$
or
$x_v \geq 1/2 \Rightarrow y_v = 1$

Let z be the OPTimum solution VC:ILP

$$\sum_v y_v \leq 2 \sum_v x_v \quad \& \qquad \sum_v z_v \geq \sum_v x_v$$

$$\leq 2 \sum_v z_v \quad \&$$

y is a 2-approximate solution for VC

f-approximate soln for SC

f = maximum number of sets that an element can occur in

I + Greedy (last lecture)

$O(\min(f, \log n))$ approximation for set cover

SCHEDULING

Given :- 1) set J of n jobs

2) set M of m machines

3) Job j has processing time p_j, j can be scheduled on machines in $M_j \subseteq M$

Goal :- Partition $\langle X_1, X_2, \ldots, X_n \rangle$ s.t. each job j is assigned to a machine in M_j that minimizes $\max_{i \in [n]} \sum_{j \in X_i} p_j$

Constructing the ILP:

x_{ij} = processing time of job j on machine

$i \in M_j$. $\qquad x_{ij} \in \{0, p_j\}$.

L = makespan.

GS: ILP.

$$\min L$$

L upper-bounds processing time on machines

$$\sum_{j : i \in M_j} x_{ij} \leq L. \quad \forall i \in M$$

Every job is completed.

$$\sum_{i : i \in M_j} x_{ij} = p_j \quad \forall j \in J.$$

$$x_{ij} \in \{0, p_j\}$$

GS: LP.

$$x_{ij} \geq 0$$

ALG

1) Solve GS:LP & let (x, L) be the optimum solution

2) Round (x, L) to an integer solution feasible for GS:ILP

How to choose good lower-bound

Let L^* be the opt solution to the GS:ILP

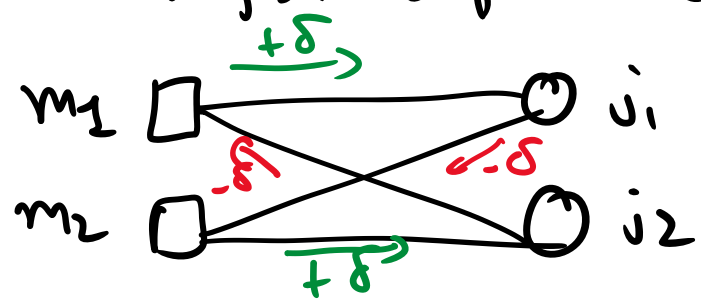
$$L^* \geq L \qquad (\text{GS:ILP} \subseteq \text{GS:LP})$$

$$L^* \geq L \qquad (GS{:}ILP \subseteq GS{:}LP)$$

$$L^* \geq \max_j p_j$$

Rounding

1) "Make the fractional solution as integral as possible"



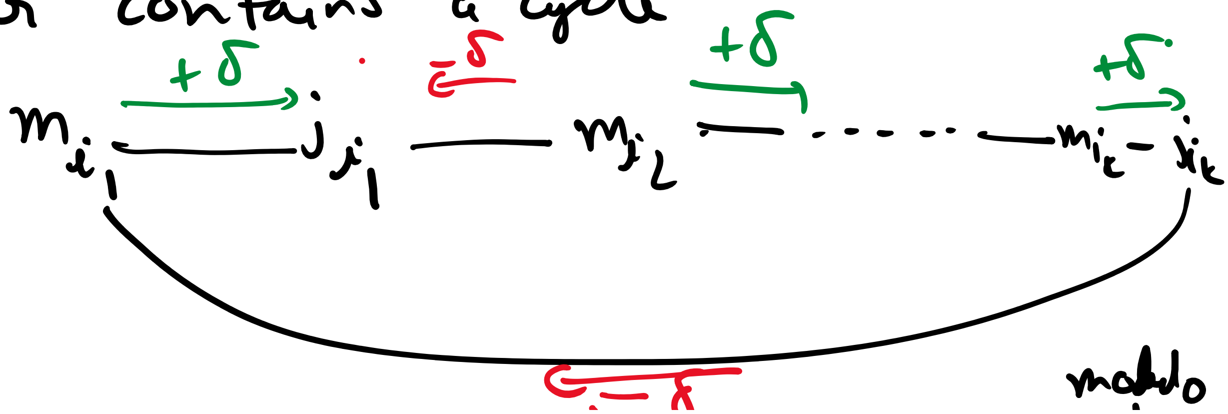
$$\delta = \min(x_{12}, x_{21})$$

Construct a bipartite graph

$$G = (M \cup J, E)$$

$$(m, j) \in E \text{ if } x_{m,j} > 0$$

If G contains a cycle



modulo

1) set $\delta = \min_\ell (x_{i_\ell, j_{i_{\ell+1}}})$ modulo k index

2) increase $x_{i_\ell j_\ell}$ by $\delta \uparrow$

3) decrease $x_{i_\ell j_{\ell+1}}$ by $\delta \downarrow$

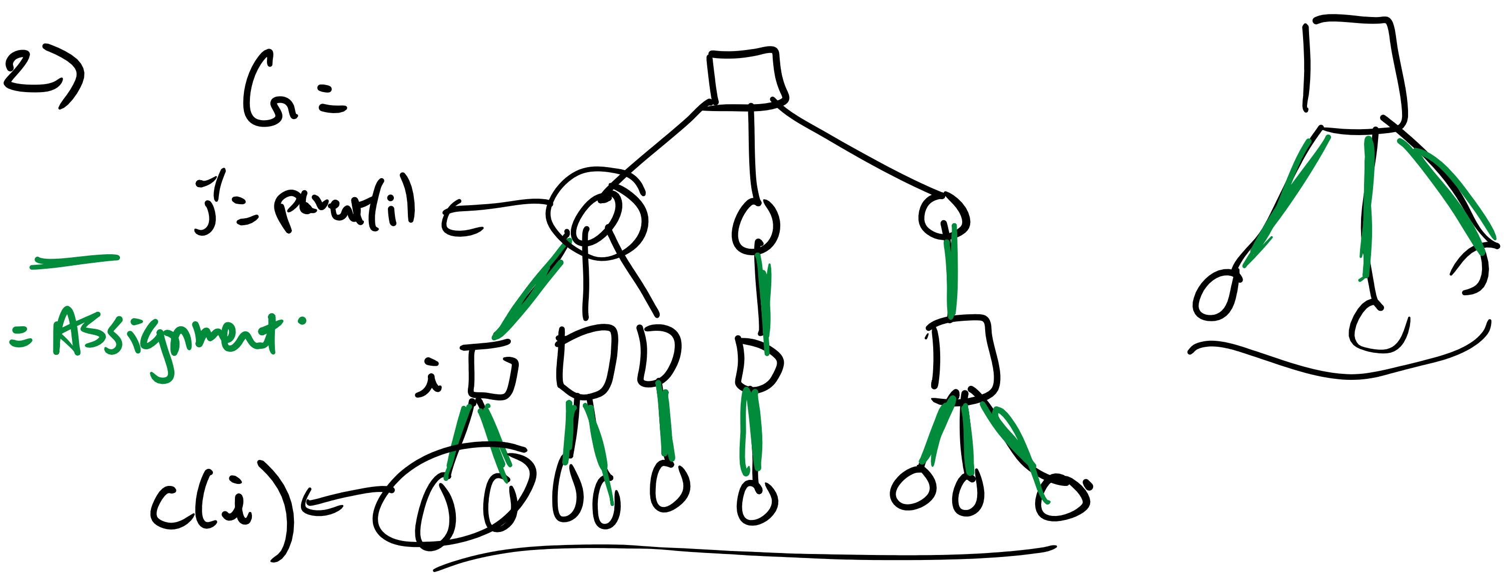
$\downarrow$

Eliminate cycle C from G.

O(nm) many cycle-eliminations

G is <u>acyclic</u>.

2) G =



a) Assign each leaf job to the parent machine.

b) Assign each internal job arbitrarily to a child-machine.

Pick a machine i ; $c(i)$ = child jobs of i in G

$$\sum_{j\in X_i} p_j = \underline{\sum_{j\in C(i)} p_j} + \underline{p_{j'}}$$

j' = parent(i) in G.

$$\leq L^* + L^*$$

$$= 2\underline{L^*}.$$

2-approximation alg for Generalized schedu

Remark: → GS: LP

$\underline{i \in M_j}$.

Randomized Rounding (next lecture)