NP Hard Review

P → polynomial time solvable, NP → poly time verifiable

Reductions. $A \le_p B$ A is poly time reducible to B.

→ A is easier than B

Review from EECS 376: Reductions

- Goal: Show problem A is not (much) harder than problem B *without solving either one.*
- $A \le_P B$ *"Problem A is polynomial-time reducible to Problem B"*
  - There is a poly-time ***reduction algorithm*** f.
  - Given instance X of problem A, $f(X)$ is an instance of problem B.
  - X is a "yes" instance of A iff $f(X)$ is a "yes" instance of B.
- E.g.,
  - BIPARTITE-MATCHING $\le_P$ MAXFLOW
    "Is there a perfect matching?" "Is the max flow equal to n?"
- B is ***NP-hard*** if for ***every*** problem $A \in NP$, $A \le_P B$.
- B is ***NP-complete*** if $B \in NP$ and B is NP-hard.

Claim Integer Linear Programming

→ max $\sum_i c_i x_i$ subject to $\sum a_{ij} x_i \le b_j \quad \forall\, 1 \le j \le m$

→ $\forall x_i \in \mathbb{R}$ or $x_i \in \mathbb{Z}$

$\Phi = C_1 \wedge \dots \wedge C_m$

$C_i = L_{i_1} \vee L_{i_2} \vee L_{i_3}$

L_{i_j} = either x_j or $\overline{x_j}$

Find assignment to $x_1 \dots$ so Φ = true

Thm 3-Sat $\le_p$ Int-lin-prog

Variables $x_1, \dots, x_n$. $x_j = \begin{cases} 0 & \text{false} \\ 1 & \text{true} \end{cases}$

max num clauses true

$\forall C_i \quad l_{i_1} + l_{i_2} + l_{i_3} \ge 1$ ← some is true.

if $l_{i_1} = x_j$ done

if $l_{i_1} = \overline{x_j}$ then let $l_{i_1} = 1 - x_j$

$0 \le x_j \le 1$

want $x_j \in \mathbb{Z}$ → if remove this $x_i = 0.5$ is soln

Doesn't actually matter → if all const satisfied then done!

⇒ need $x_j \in \mathbb{Z}$.

So done here!

Decision problems → Optimization problems

- CLIQUE(G): Find ***maximum*** k s.t. G contains a k-clique
- INDEPSET(G): Find ***maximum*** k s.t. G contains a k-indep. set
- VERTEXCOVER(G): Find ***minimum*** k s.t. G contains a k-vertex cover.

- Let $k_{OPT}(G)$ be the optimum value for an instance G.
- Let $k_A(G)$ be an answer returned by algorithm A, given G.
- ***Definition.*** *A is a C-approximation* of problem [blah] if
  - $k_A(G) \le C \cdot k_{OPT}(G)$ (for minimization problems)
  - $k_A(G) \ge k_{OPT}(G)/C$ (for maximization problems)

- An approximation algorithm template:
  - 1. Express your problem as an instance of (ILP)
  - 2. "Relax" all integrality constraints → (LP)
  - 3. Let x^* be optimum solution to (LP)
  - 4. "Round" x^* to integer vector $\hat{x}$. Measure difference in objective quality between x^* and $\hat{x}$.

e.g.

Weighted Vertex Cover

- Input: $G(V, E)$ and $w: V \to \mathbb{R}^+$.
- Output: V' minimizing $w(V')$ such that for all $\{u, v\} \in E$, $V' \cap \{u, v\} \neq \emptyset$.

Variables

$x_1, \dots, x_n$ for each vertex,

x_i represents choosing vertex v_i.

$\min \sum x_i w(v_i)$ } obj

$\forall (i,j) \in E \quad x_i + x_j \geq 1$ → each edge covered

$0 \leq x_i \leq 1$, $x_i \in \mathbb{Z}$

let x^* is ~~opt~~ soln to LP (relax $x_i \in \mathbb{Z}$)

Obs for $(i,j) \in E$ either x_i or $x_j \geq 0.5$

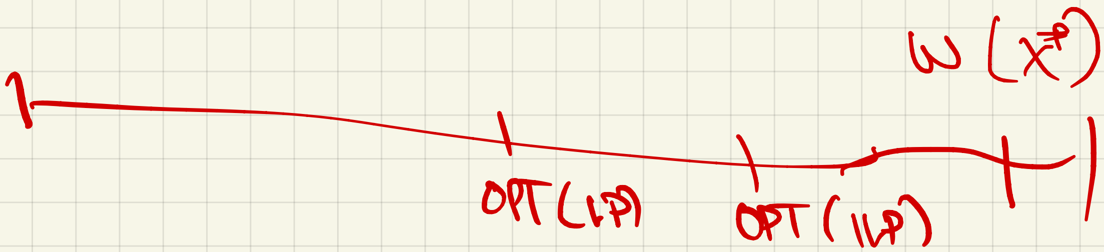
→ call rounding by

$$\hat{x} = \begin{cases} \hat{x}_i = 0 & \text{if } x_i^* < 0.5 \\ \hat{x}_i = 1 & x_i^* \geq 0.5 \end{cases}$$

$\Rightarrow w(\hat{x}) \leq 2 \cdot w(x^*) = 2 \text{ opt}(LP)$ more const.

$\leq 2 \cdot OPT(ILP)$

$\Rightarrow \frac{1}{2} w(\hat{x}) \leq OPT(ILP)$ → 2 approx!



Weighted Set Cover

- **Input:** Set system $(X, \mathcal{S})$ where $\mathcal{S} = (S_1, S_2, \dots, S_m)$ and $S_i \subseteq X$, and a weight function $w: \mathcal{S} \to \mathbb{R}^+$.
- **Output:** $T \subseteq \mathcal{S}$ minimizing $w(T)$ such that $\bigcup_{S_i \in T} w(S_i) = X$

let $x_1, \dots, x_m$ variables. $x_i = \begin{cases} 0 & S_i \notin \text{soln} \\ 1 & S_i \in \text{soln} \end{cases}$

min $\sum w(i) x_i$

$$\left.\begin{aligned} &\forall j \in X \quad \sum_{j \in S_i} x_i \geq 1 \\ &x_i \in [0,1] \cap \mathbb{Z} \end{aligned}\right\} \to \text{constr.}$$

make LP x^* is opt for LP

let. $\hat{x}_i = \begin{cases} 0 & \text{else} \\ 1 & x_i^* \geq 1/m \end{cases}$

$\Rightarrow w(\hat{x}) \leq m \cdot w(x^*) \cdots\to$ m approximation

$\hookrightarrow$ this is a lot.

let T_0 be fuzzy soln

$\hookrightarrow$ $S_i \in T_0$ with prob x_i

treat this as p

Cost of T_0 $\to E(w(T_0)) = \sum E(I_j) w_j = \sum x_j^* w_j$

I_j for j in soln

$\hookrightarrow$ optimum of LP!

Coverage $j \in X$ $P(j \text{ is uncovered by } T_0)$

$$= \prod_{j \in S_i} (1 - x_i^*) \leq \prod_{j \in S_i} e^{-x_i^*} = e^{-\sum_i x_i^*} \leq \frac{1}{e}$$

Randomized Rounding

x^*

Repeat $\ell \in 0, \dots, k-1$

T_ℓ = include i with prob x_i^*

Return

$T_0 \cup T_1 \cup \dots \cup T_{k-1}$

$E[\text{cost}] = k \cdot OPT(LP) \le k \cdot OPT(ILP)$

lin

union bd

$P(\text{point is uncovered}) = |X| e^{-k}$ → even ind

→ $k = \ln|K| + c$ → for chosen upper bd here

i.e. $|x| e^{-k} < \frac{1}{2}$ → c is small → just solve

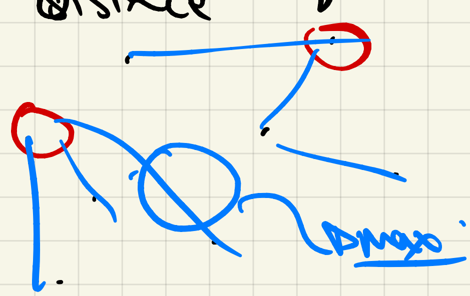
K-Center 1

have a bunch of points with budget k.

→ can choose k points min furthest distance to chosen point.

So, $P = (p_1, \dots, p_n)$ set of points

& metric $d: P \times P \to \mathbb{R}_{\geq 0}$.



Greedy appr

→ Start with arb $C = \{i\}$

while $|C| \leq k$

$j = \arg\max_j (d(p_j, C))$ → farthest away from center

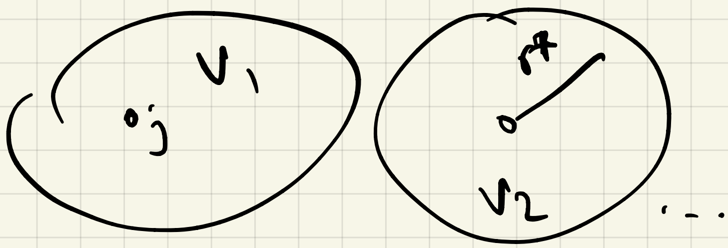
$C = C \cup \{j\}$

:

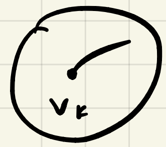
→ Claim C is a 2-approx

What is C^* (optimal)? value r^*

↳ it partitions P into regions (closest elt in C)



↝ r^* balls around elts of C^* cover space.



Case 1) C has 1 pt in each of $V_1, \dots, V_k$

$\Rightarrow$ by triangle inequality, dist $\leq 2 \cdot r^*$

$\Rightarrow$ $\text{val}(C) \leq 2 \cdot r^*$

Case 2) $\exists$ 2 points in some V_j → in fact the second we double down we have a 2 approx → call seq of choices $i_1, \dots, \underbrace{i_{f-1}\ i_f}_{\text{in same } V_j}$