Lecture 2:

problem 2.2. MAX - CUT

Alg: • start with arb. partition

• if $\exists$ a vertex s.t. flipping increases cut size, flip it.

Show: • terminates in poly-time ?

• is a $\frac{1}{2}$-approx

pf: let m be the size of the max-cut (in edges)

~~• Suppose Alg terminates with $\frac{m}{2} - 1$ edges.~~

~~$\Rightarrow$ $\frac{m}{2}+1$ edges of the max cut we find in the current cut.~~

~~MAX-CUT(I)~~

$$OPT(I) \le \sum_{u \in V} d(u) \le 2A(I)$$

Claim: $\ge \frac{1}{2}$ the edges incident on each node are in the cut generated by Alg.

pf: otherwise, flip node to get better cut.

$$\Rightarrow \sum_{u \in V} d(u) \le 2A(I)$$

Claim: $OPT(I) \le \sum_{u \in V} d(u)$

What about Weighted MAX-CUT?

Alg: • pick a side uniformly at random for each vertex

pf: prob edge e is in the cut $= \frac{1}{2}$.

$$E[A(I)] = \sum_{e \in E} \frac{1}{2} \cdot w_e = \frac{1}{2} w(E)$$

but. $OPT(I) \leq w(E)$.

• can de-randomize this:

Alg: • process in arb. order. (vertices)

• place vertex on side ~~those~~ that gives more cut edges.

pf. (say process $v_0 v_1 \dots v_n$

define $E_j = \{ v_i v_j \in E \mid i < j \}$ gives a partition of E.

• at least $\frac{1}{2} w(E_j)$ is in the cut $\forall j$

$$\Rightarrow A(I) \geq \frac{1}{2} \sum w(E_j) = \frac{1}{2} w(E)$$

• KNAPSACK FPTAS.

• NP-hard probs don't admit poly-time algs.

$$\text{poly}(|I|)$$

• pseudo-poly-time Alg $\text{poly}(|I_u|)$ ← in unary.

KNAPSACK: S - set of objects, $|S| = n$; $\text{size}(s)$ - sizes; $\text{profit}(s)$ - profits; knapsack capacity B.

• let P be profit of most profitable object.

$$P \triangleq \max_{a \in S} \text{profit}(a)$$

• nP is an upper bound on the profit possible.

• $S(i,p) \triangleq$ a subset of $\{a_1 \cdots a_i\}$ whose profit is exactly p and whose $\underline{\underline{\text{size}}}$ is minimized.

• $A(i,p) \triangleq \text{size}(S(i,p))$ (∞ if no such set exists)

• $A(1,p)$ for $p \in \{0, 1, \cdots nP\}$ easy.

$$A(i+1,p) = \begin{cases} \min\{A(i,p), \text{size}(a_{i+1}) + A(i, p - \text{profit}(a_{i+1}))\} & \text{if } \text{profit}(a_{i+1}) \le p \\ A(i,p) & \text{o/w.} \end{cases}$$

Ans: $\max_{A(n,p) \le B} p$

time: $O(n^2 P)$; i ranges $1 \cdots n$, p ranges $1 \cdots nP$

- if P were $poly(n, \frac{1}{\varepsilon})$ we would be good.

- idea : look at the profits with "fuzzy" glasses

Alg: - Given $\varepsilon > 0$, let $k = \frac{\varepsilon P}{n}$ ← fuzzy glasses.

- define $profit'(a) = \left\lfloor \frac{profit(a)}{k} \right\rfloor$

- solve dynamic prog alg as before to get S',

- output S'

Claim. $profit(S') \geq (1-\varepsilon)\, OPT$

pf. - $\forall a$ $profit(a) \leq profit'(a) k + k$ and $profit(a) \geq profit'(a) k$

$\Rightarrow profit(O) \leq profit'(O) k + nk$

since $profit'(S')$ maximizes

$\Rightarrow profit(O) \leq profit'(S') k + nk$

$$(1-\varepsilon) OPT \underset{\text{rewrite}}{\leq} OPT - \varepsilon P = profit(O) - nk \leq profit'(S') k \leq profit(S')$$

def k

def profit'

since $OPT \geq P$

max profit is $\left\lfloor \frac{p}{k} \right\rfloor$ $\frac{n}{\varepsilon}$

new run time

$$O\left(n^2 \frac{n}{\varepsilon}\right) = O\left(n^3 \frac{1}{\varepsilon}\right)$$

Strong NP-hardness. (not many problems with "FPTAS")

NP complete problem P

$$Q \xrightarrow{\text{poly time reduction}} P$$

$$Q \text{ iff } P.$$

strong NP-complete P ← most problems are strongly NP-hard.

$$Q \xrightarrow{\text{poly time, all numbers for } P \text{ written in unary}} P$$

$$Q \text{ iff } P$$

- Strong NP-hard ⇒ no ~~FPTAS~~ pseudo poly time alg.
  - why?
- FPTAS ⇒ pseudo poly alg.

Thm: p - poly. Π is NP-hard s.t. f_Π - obj func, is integer.
& $OPT(I) < p(|I_u|)$

if Π admits FPTAS it also admits pseudo-poly.

pf. ∘ suppose FPTAS run time is $g\left(|I|, \frac{1}{\varepsilon}\right)$

∘ set $\varepsilon = \frac{1}{p(|I_u|)}$

- $OPT(I) \le A(I) \le (1+\varepsilon) OPT(I) = OPT(I) + \varepsilon OPT(I) < OPT(I) + 1.$
- so, found optimal in $g\left(|I|, \frac{1}{p(|I_u|)}\right)$ time - pseudo poly.