Lecture 3

Sunday, December 4, 2022 12:44

Vertex Cover

In: G

TASK: Find $S \subseteq V(G)$ st $\forall uv \in E(G)$ $\{u,v\} \cap S \neq \emptyset$

Parameterized by OPT:

In: G, k

Q: $\exists$ vertex cover of size $\leq k$?

naive: $n^k \cdot m$

Can we do BETTER?

$2^k n^{O(1)}$ ← Fixed PARAMETER TRACTABLE $f(k) n^c$ for c independent of k.

Cute randomized algorithm:

```
S = ∅
while G-S has at least one edge
    pick uv ∈ E(G-S) u.a.r.
    pick s ∈ {u,v} u.a.r
    S = S ∪ {s}
output S
```

- ALG always runs in polynomial (linear!) time
- S is always a vertex cover (why?)
- What is $\Pr[S \text{ is an OPTIMAL v.c.}]$?
  - Let OPT be some fixed optimal vertex cover, suppose $|OPT| \leq k$.
  - Initially $S \subseteq OPT$.

    Claim: in each round, if $S \subseteq OPT$ then $\Pr[s \in OPT] \geq 1/2$

    Proof: uv has ≥ 1 endpoint in $OPT - S$!
  - $|S|$ increases by 1 in every round. If $s \in OPT$ in every round then $S = OPT$!

    ↑ happens with probability $\geq (1/2)^k = 1/2^k$
  - S is optimal with prob $\geq 1/2^k$

(repeat 2^k times and keep smallest soln to get success prob $1 - \left(1 - \frac{1}{2^k}\right)^{2^k} \geq 1 - 1/e$)

Problem for tutorial:

① Show that the number of inclusion minimal vertex covers of size $\leq k$ is at most 2^k

This algorithm is also a factor 2 approximation algorithm!

- For a graph G, define X_G to be the random variable returning the size of the set S output by the algorithm
- For integers k, n define

$$X_{k,n} = \max_G E[X_G]$$

where max is taken over all graphs with $\leq n$ vertices and vertex cover of size $\leq k$

- In each step of ALG, with probability $\geq 1/2$, the vertex s selected by ALG is contained in an optimal v.c. of $G - S$

$$X_{n,k} = E[X_G] \leq 1 + \frac{1}{2} E[X_{G-v} \mid v \in OPT] + \frac{1}{2} E[X_{G-v} \mid v \notin OPT]$$

$$\leq \frac{1}{2} X_{n,k-1} + \frac{1}{2} X_{n,k}$$

Hence $X_{n,k} \leq 2 + E[X_{n,k-1}] \leq 2k$!

Tutorial: use Markov inequality to show that $\Pr[|S| \leq 2 \cdot |OPT|] \geq \Omega(1/|OPT|)$

Feedback Vertex Set:

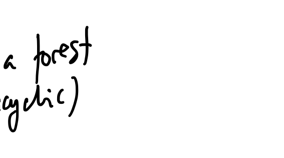
In: G

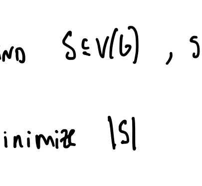
TASK: Find $S \subseteq V(G)$, s.t $G - S$ is a forest (acyclic)

Minimize $|S|$

- Will see 4-APPX & 4^k ALG.

Will work with multi-graphs





Lemma 1: Let G be a multigraph, v be a vertex of degree ≤ 1. Then:

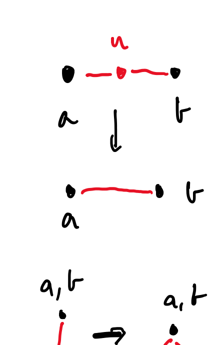
- Min FVS of $G - v \leq$ Min FVS of G
- $\forall S \subseteq V(G) - \{v\}$ S is an FVS of G ⇅ S is an FVS of $G - v$

Pf: why?

Lemma 2: If G has a vertex v with a self loop then

- Every FVS S of G contains v
- For every S containing v: S is an FVS of G ⟺ $S - v$ is an FVS of $G - v$

Lemma 3: Let G be a (multi) graph with no self loops, u be a vertex of degree 2, a and b be the neighbors of u ($a = b$ is possible!) G' be the graph obtained by adding an edge ab to $G - u$



Then:

- Every FVS S of G s.t $u \notin S$ is an FVS of G'
- Every FVS S of G' is an FVS of G

Pf: ?? [where do we use that G has no self loops?]

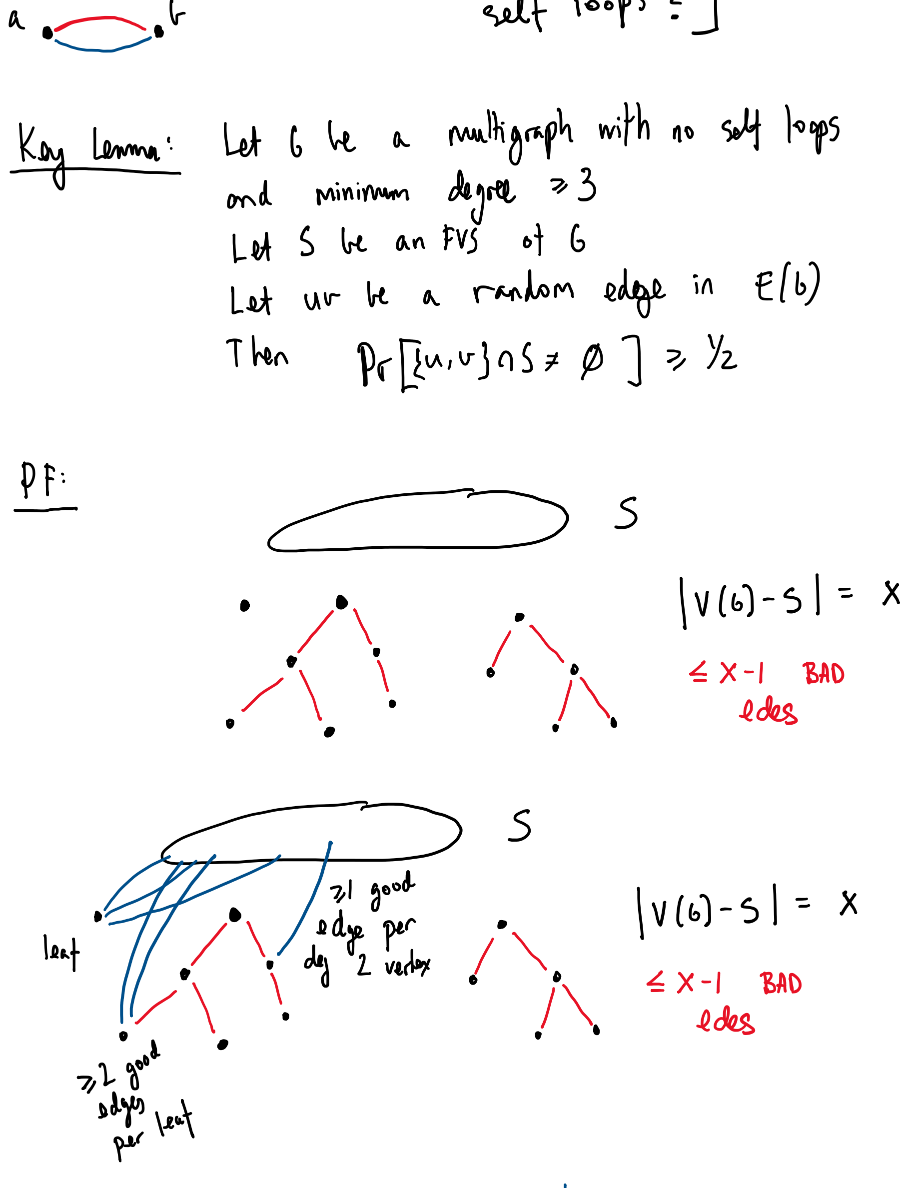
Key Lemma: Let G be a multigraph with no self loops and minimum degree ≥ 3

Let S be an FVS of G

Let uv be a random edge in $E(G)$

Then $\Pr[\{u,v\} \cap S \neq \emptyset] \geq 1/2$

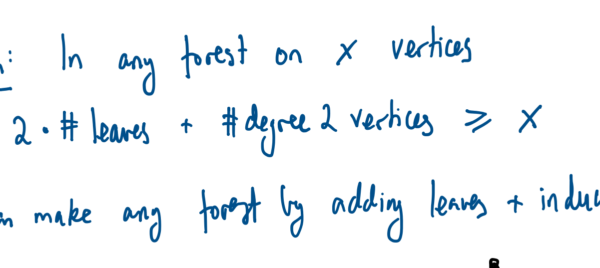
PF:



Claim: In any forest on x vertices

$2 \cdot \#\text{leaves} + \#\text{degree 2 vertices} \geq x$

Pf: Can make any forest by adding leaves + induction



```
ALG(G)
    if G is acyclic return ∅
①   if ∃ self loop v    RETURN ALG(G-v) ∪ {v}
①   if ∃ degree ≤1 vertex v    RETURN ALG(G-v)
③   if ∃ degree ≤2 vertex v    Return ALG(G')
                                 (G' = G/v)
    pick ede uv ∈ E(G) u.a.r
    pick s ∈ {u,v} u.a.r.
    RETURN ALG(G-s) ∪ {s}
```

Lemma: ALG always returns FVS of G

Pf: LEMMAS 1,2,3 + induction

Lemma 2: If G has an FVS of size k then $\Pr[\text{Alg returns OPTIMAL FVS}] \geq 1/4^k$

Pf: Induction on n + Lemmas 1,2,3 + KEY.

→ poly time alg with success probability $1/4^k$

↓

$4^k \text{poly}(n)$ time alg with success probability $\geq 1 - 1/e$!

SIMILAR ANALISIS AS FOR V.C. SHOWS THAT ALG IS A POLY TIME 4-APPROXIMATION IN EXPECTATION

$$X_{n,k} \leq \max \begin{cases} 1 + X_{n,k-1} \\ X_{n-1,k} \\ 1 + \frac{1}{4} X_{n,k-1} + \frac{3}{4} X_{n-1,k} \end{cases}$$

which solves to $X_{n,k} \leq 4k$

Have seen APPROXIMATION AND PARAMETERIZED ALGS.

EXACT: Brute force is $2^n \text{poly}(n)$

CAN WE DO BETTER??

- For V.C. ALG:
- Given G, k

  pick $t \leq k$ "cleverly"

  pick $Z \subseteq V(G)$ u.a.r

  Add Z to solution

  Solve $G - Z$ using FPT alg on $(G - Z, k - t)$ in time 2^{k-t}

GENERIC SCHEME – ALSO WORKS FOR FVS:

- For FVS ALG:
- Given G, k

  pick $t \leq k$ "cleverly"

  pick $Z \subseteq V(G)$ u.a.r

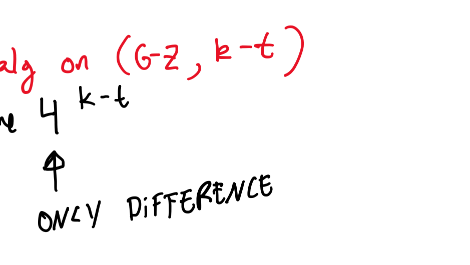
  Add Z to solution

  Solve $G - Z$ using FPT alg on $(G - Z, k - t)$ in time 4^{k-t}

  ↑ ONLY DIFFERENCE

running time: 2^{k-t} for VC, 4^{k-t} for FVS $\}$ c^{k-t}

Success probability: $\geq \dfrac{\binom{k}{t}}{\binom{n}{t}}$



repeat $\dfrac{\binom{n}{t}}{\binom{k}{t}}$ times for success prob $1 - 1/e$

running time: $\dfrac{\binom{n}{t}}{\binom{k}{t}} \cdot c^{k-t}$

↓

$$\min_{t \leq k} \frac{\binom{n}{t}}{\binom{k}{t}} \cdot c^{k-t}$$

↓

$$\max_{k \leq n} \min_{t \leq k} \frac{\binom{n}{t}}{\binom{k}{t}} \cdot c^{k-t} \leq \left(2 - \frac{1}{c}\right)^n \cdot \text{poly}(n)$$

↑ math

→ 1.5^n alg for V.C.

1.75^n alg for FVS