Intro, Vertex Cover

Course Staff: Me, Aditya Krishnan

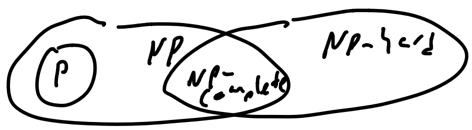
Approximation Algorithms: What and Why

P: Solvable in poly time

NP: Verifiable in poly time

NP-hard: Problems where all problems in NP reduce to in poly time

NP-complete: in NP and NP-hard



Given NP-hard optimization problem, what do we want that NP-hardness prevents?

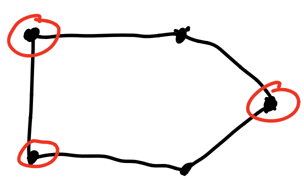
1. Find the optimal solution: approximation algs
2. In polynomial time: allow slower algs
3. For every instance: "average-case", special inputs

Problem consists of:

1) Description of inputs/instances
2) Description of feasible solutions for input
3) Objective function

Example: Vertex Cover

- Input: Graph $G = (V, E)$
- Feasible solutions: $V' \subseteq V$ s.t. $e \cap V' \neq \emptyset \ \forall e \in E$
- Objective: minimize $|V'|$



Def: The optimal solution is the feasible solution with the best objective value

Def: A some problem, I instance of A

- ALG: some algorithm for A
- OPT(I): objective value of optimal solution for I
- ALG(I): objective value of solution returned by ALG on I

ALG is an α-approximation if:

- always returns a feasible solution
- runs in polytime
- $$\frac{ALG(I)}{OPT(I)} \leq \alpha \quad \forall \text{ instances } I \text{ of } A \quad \text{(min problem)}$$

$$\frac{ALG(I)}{OPT(I)} \geq \alpha \quad \forall \text{ instances } I \text{ of } A \quad \text{(max problem)}$$

$$\frac{OPT(I)}{ALG(I)} \leq \alpha$$

α is the approximation ratio or approximation factor

- "Fine-grained complexity"
- Focus is to design new techniques

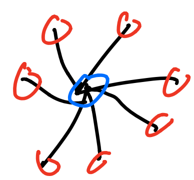
A_1, A_2 problems.

A_1: design a 2-approximation

A_2: Assuming $P \neq NP$, no α-approx for $\alpha < 10$

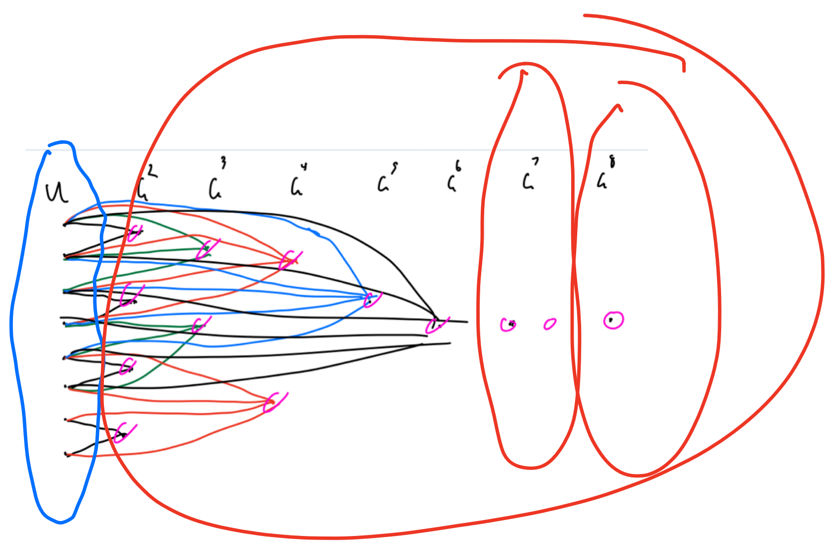
Approximating Vertex Cover

Idea 1: Arbitrarily add vertices until finished



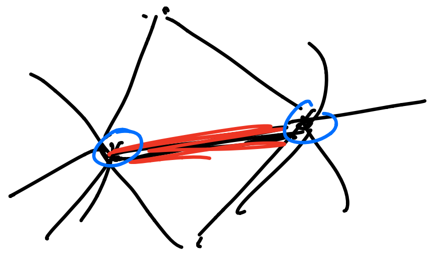
$$\alpha \geq \frac{n-1}{1}$$

Idea 2: Greedy algorithm



$$\frac{\text{ALG: } n}{\text{OPT: } \frac{n}{\log n}} = \log n$$

Better Algorithm:



- $S = \emptyset$
- while $E \neq \emptyset$ {
    - Let $\{u,v\}$ be arbitrary edge
    - $S \leftarrow S \cup \{u,v\}$
    - Remove u,v, all incident edges from G

  }

Poly time: ✓

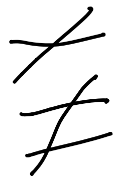
Lemma: S feasible (S a vertex cover)

Pf: ✓

Def: $M \subseteq E$ is a matching if no two edges in M share an endpoint: $e \cap e' = \emptyset$ $\forall e, e' \in M$ with $e \neq e'$

Lemma: Let M be a matching, and S be a vertex cover. Then $|S| \geq |M|$

Pf:



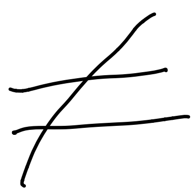
Thm: Algorithm is a 2-approx.

Pf: polytime, feasible: ✓

Let S^* optimal vertex cover.

WTS: $|S| \leq 2|S^*|$

Edges found by alg: matching M



$|S| = 2|M| \leq 2|S^*|$

by lemma

Best alg: $2 - \frac{1}{\sqrt{\log n}}$