Some languages in P cannot be NP-Complete.

Surprisingly, off-tape reductions of NP-hard for all O(\text{time complexity}).

For others, even current, version is

2 - Complement for Max-Cut
2 - Complement for Min-Vertex-Cover

Has a poly-time algorithm?

Approximation version of the problem

For some NP-Complete languages, the

Size of Cut

\text{Max-Cut} : \text{Find} \quad \text{max} \quad \text{cut} \quad \text{size} \quad \frac{1}{2} \quad \text{max possible}

\text{Min-Vertex-Cover} : \text{Find} \quad \text{min} \quad \text{cover} \quad \text{size} \leq \frac{1}{2} \quad \text{max possible} \quad \# \text{of clauses}

\text{Ex: Max-3-SAT} : \text{Find an assignment that}

\text{problems} \quad \exists \text{A, B (a, b)} \quad \exists \text{A, B (a, b)}

\text{Approximation} \quad \text{on input} \quad \text{output X, Y s.t.} \quad \frac{1}{2} \\text{sat}

\text{vertex cover, find cut with most edges}

\text{Ex: Satisfy max} \# \text{of clauses in SAT, smallest}

\text{problems} \quad \text{on input} \quad \text{output} \quad \frac{1}{2} \\text{sat}

\text{Ex: Find satisfiable, vertex cover of size K}

\text{problems} \quad \text{on input} \quad \text{output} \quad \frac{1}{2} \\\text{sat}

\text{Ex: Decade 3SAT, decide vertex cover, decision cut}

11/29

Randomized Computation

CS 1010 Lecture 23
2-Aprox for min-vertex-cover:

on input \( G \)

repeat until break

- Find an unmarked edge that does not share a vertex with a marked edge
- if such an edge exists, mark it.
- else, break

output the vertices that are the endpoints of all the marked edges.

Analysis:
- runtime: iterates at most \( |E| \) times, polytime \( \checkmark \)
- vertex cover: Every edge is either marked or adjacent to a marked edge. All the vertices on either side of marked edges are in the cover, so both marked and unmarked edges are covered.
- 2-approx: let \( Y \) be our approximation, \( H \) be a set of half the size: one endpoint of every marked edge. Let \( Z \) be the min vertex cover.
- \( |Y| = 2|H| \)
- \( |Z| \geq |H| \) \( \leq \) every marked edge must have a node in \( Z \)

so, \( |Y| \leq 2|Z| \)
$2$-Approx for Max-Cut.

On input $\langle G \rangle$

Let $S = \emptyset$, $T = V(G)$

Repeat until break $\emptyset$

* If moving some $v$ to the other side of the cut, move $v$.
* Else, break.

Analysis:

* Runtime: Each iteration is polytime, and will halt in fewer than $|E|$ iterations (bound on cut size), so it is polytime.

* Outputs a cut by construction.

* $2$-approx: will show that the size of the cut is $\geq \frac{1}{2} |E|$

For every $v$, let $u_1, ..., u_k$ be its neighbors, $\geq \frac{1}{2}$ of them are on the other partition.

$\# \text{ of cut edges} = \sum_{v \in V} \# \text{ of cut edges adj to } v$

$\geq \sum_{v \in V} \# \text{ of edges adj to } v$

$\geq \frac{2}{2} |E| = \frac{1}{2} |E|$
FACT: CLIQUE is hard to approximate within some constant $\epsilon$. Let us see that it is hard to approximate in any $NP$-hard to decide between 2 possible inputs:

yes: $\langle G, k \rangle$ w/ a clique of size $\leq k$
no: $\langle G, k \rangle$ w/ a clique of size $\leq \frac{1}{2} k$

Product graph: $G^2 = \langle \mathcal{V} \times \mathcal{V}, \mathcal{E}^2 \rangle$ if $(u_1, u_2), (v_1, v_2) \in \mathcal{E}$ and $(u_2, v_2) \in \mathcal{E}$

Can we deciding the product graph to shrink the $NP$-hard decision problem -> contradiction

Cut = $\{ \langle G, k \rangle \mid \exists$ a $k$-cut of $G \}$
a cut is a partition of $V$. Its size is the number of edges from one side of the partition to the other.