Lecture 15: \([\text{NP Completeness}] \text{ and } [\text{Approximations}]\)

- P vs NP

- blah "unless \( P = NP \)"

- unless something weird happens.

- NP-complete ("hardest problem in NP")

\( Q \) is NP-complete

\[ f: Q \in \text{NP} \]

(2) \( \exists \) another NP-complete problem \( Q' \) s.t.

\[ Q \in P \implies Q' \in P \]

(3) 3SAT in NP-complete

if \( Q \in P \) then NP = P
1) Independent Set / Stable Set

- **Input**: graph $G$, number $K$
- **Question**: does there exist an Independent Set in $G$ of size $\geq K$?

Claim: Independent Set is NP-complete

1. Independent Set is in NP (✓)
2. "reduce an NP-hard problem to Independent Set"
   - if Independent Set is in $P$ $\implies$ 3SAT is in $P$
3SAT: \[(x_1 \lor x_3 \lor x_7) \land (x_5 \lor \overline{x_{n-1}} \lor \overline{x_1}) \land \cdots \land (x_1 \lor \overline{x_3} \lor x_5)\]

Want to solve 3SAT using an edge for 1st SET

\[G = \text{diagram}\]

does \(\exists\) an und. set \(S\) on \(K = (\#\text{clauses})\)
**Theorem**

**Vertex Cover** is NP-complete

Let \( G = (V, E) \) be a graph. A **vertex cover** \( C \) is a subset of \( V \) such that every edge \( (u, v) \) in \( E \) has at least one of its endpoints in \( C \).

(Cover edges using vertices)

**Problem**

Given \( G \) and an integer \( K \), does \( G \) contain a vertex cover \( C \) such that \( |C| \leq K \)?
1. VC is NP
2. if VC is P ⇒ Ind Set is in P. (⇒ "which would be weird")

Claim: Graph G, S in an Ind set ⇔ V \ S is a vertex cover.

(G, K) in a YES instance of VC
⇔ (G, n-K) in a YES instance of IndSet
Load Balancing

Input: Jobs

"processing" sizes:

\[ \begin{array}{cccc}
1 & 2 & \cdots & n \\
P_1 & P_2 & \cdots & P_n \\
\end{array} \]

\[ \text{\( n \) machines} \]

Max load over all machines

Minimize
Q: Solve Makespan Minimization fast?
**Thm 1**: $MM$ is NP-complete

- If $P \in P \Rightarrow MM \in P$

**Thm 2**: $\exists$ approx to $MM$

**MM instance**: $(m, p_1, p_2, \ldots, p_n, K)$
- $m$: #machines
- $p_i$: sizes
- $K$: makespan

**Thm**:
1. $MM \in NP$  (hint = assignment)
2. If $MM \in P \Rightarrow \text{Partition} \in P$

**Partition**: given numbers $a_1, a_2, \ldots, a_L \geq 0$

- Does there exist $S \subseteq \{1, 2, \ldots, L\}$
- $\sum_{i \in S} a_i \leq \sum_{i \notin S} a_i$

**Thm**: $\text{Partition is NP-complete}$

\[ m=2 \]
\[ P_i = a_i \]
\[ K = \frac{\sum ai}{2} \]
Approximation Algorithm for Makespan Min

\[ 2 \quad 1 \quad 3 \quad 4 \quad 7 \quad 2 \quad 4 \quad 5 \quad 7 \quad m = 3 \]

Greedy:
- Sort jobs in decreasing order of size
- For \( j = 1 \) to \( n \)
  - If \( n \) jobs
    - Put job \( j \) on least loaded machine so far

\[
\begin{array}{c c c c}
7 & 4 & 5 \\
\hline
7 & 4 & 2 \\
\hline
2 & 1 & 3 \\
\end{array}
\]

Makespan of greedy = 16

\[ \left\lfloor \frac{35}{3} \right\rfloor \leq \text{OPT} \leq 14 \]
Fact: $\text{Makespan (Greedy)} \leq 2 \cdot \text{OPT}$

Proof: $\text{Greedy} = \text{last job on most loaded } m/C + \text{rest}$

$= P_{\text{last}} + (\text{rest})$

$\leq P_{\text{max}} + \frac{\sum P_j}{m}$

$\leq \text{OPT} + \text{OPT}$

$= 2 \cdot \text{OPT}$ ☺

$\text{Approx. Guarantee}$

Worst case bound on the performance of a heuristic

$\text{Greedy}$

$\leq m$

$P_1 + P_2 + \ldots + P_8$

$\leq \frac{\sum P_8}{2}$
**Thm.** Greedy $\leq (2 - \frac{1}{m}) \text{OPT}$

**Thm.** 3 examples where $\mathcal{I}$ is tight

**Thm.** Sorted Greedy $\leq (1.5) \text{OPT}$ (Easyish)

$\leq (1.5 - \frac{1}{m}) \text{OPT}$ (Easyish)

$\leq (\frac{4}{3} - \frac{1}{m}) \text{OPT}$ (Tidy)

$\leq H \cdot (1 - \frac{1}{m}) + H$