Lecture 18

1. One more NP-completeness reduction
2. Other hard problems (a quick word)
3. Approximation Algorithms
NP and co-NP

• **NP.** Decision problems for which there is a poly-time certifier.
  
  Ex. SAT, HAM-CYCLE, COMPOSITES.

• **Def.** Given a decision problem \( x \), its complement \( \bar{x} \) is the same problem with yes and no answers reversed.

• **co-NP.** Complements of decision problems in NP.
  
  Ex. TAUTOLOGY, NO-HAM-CYCLE, PRIMES.
Asymmetry of NP

• Asymmetry of NP. We only need to have short proofs of \textit{yes} instances.

Example: SAT vs. TAUTOLOGY.

– Can prove a CNF formula is satisfiable by giving such an assignment.

– How could we prove that a formula is not satisfiable?

• coNP: We only need to have short proofs of \textit{no} instances.
Approximation Algorithms
What do we do with hard problems?

- Use exponential time algorithms
- Heuristics
- Here: use a pretty good algorithm
  i.e., gives a high quality solution, but maybe not the best
Optimization problems:

• Specify valid outputs (solutions) for each input
• Specify cost function $c(y)$ for each output $y$
• Specify minimize or maximize?

• Optimal algorithm for the problem:
  For all inputs, it outputs $OPT$ such that $c(OPT)$ is min (or max) over $c(y)$ for all valid outputs $y$. 
\(\alpha\)-Approximation algorithm

Minimization:

For all inputs, algorithm outputs \(A\) such that

\[
\frac{c(A)}{c(\text{opt})} \leq \alpha
\]

Maximization:

For all inputs, algorithm outputs \(A\) such that

\[
\frac{c(\text{opt})}{c(A)} \leq \alpha
\]

\(\alpha > 1\) is the approximation ratio – can depend on \(n\)
Max- Clique

Input: An undirected graph G=(V,E) be and K >0

Problem: Find a maximal size subset C of V, that every pair of vertices in C has an edge between them?

Theorem [Lecture 17]: The decision variant Clique is NP-Complete
Mass Mailing

Say you’d like to send some message to a large list of people (e.g. all campus)

There are some available mailing-lists, however, moderator of each list charges $1 for each message sent

You’d like to find the smallest set of lists that covers all recipients
MIN SET-COVER

• **Input**: a finite set $U$ of size $m$ and a family $F$ of $n$ subsets of $U$,
such that $U = \text{union of sets in family } F$

• **Problem**: to find a set $C \subseteq F$ of minimal size which
  covers $U$, i.e $U = \text{union of sets in } C$
SET-COVER: Example
Summary

• We’ve seen some approximation algorithms:
  • For CLIQUE (ratio n/log n)
  • for SET-COVER (logarithmic ratio)
  • for VERTEX-COVER (ratio 2)

• Strange thing:
  • Some NP-complete problems hard to approximate well, others easy...
Moral

The fact that decision versions are computationally equivalent doesn’t mean that problems can be approximated within the same factor!