CS21 Lecture 22
February 26, 2024

Outline
- NP-complete problems: independent set, vertex cover, clique…
- NP-complete problems: Hamilton path and cycle, Traveling Salesperson Problem
- NP-complete problems: Subset Sum
- NP-complete problems: NAE-3-SAT, max cut

SUBSET-SUM is NP-complete

**Theorem**: the following language is NP-complete:

\[
\text{SUBSET-SUM} = \{ (S = \{ a_1, a_2, a_3, \ldots, a_k \}, B) : \text{there is a subset of } S \text{ that sums to } B \}
\]

• Proof:
  - Part 1: SUBSET-SUM ∈ NP. Proof?
  - Part 2: SUBSET-SUM is NP-hard.
    • reduce from?

We are reducing from the language:

3SAT = \{ \varphi : \varphi \text{ is a 3-CNF formula that has a satisfying assignment} \}

to the language:

\[
\text{SUBSET-SUM} = \{ (S = \{ a_1, a_2, a_3, \ldots, a_k \}, B) : \text{there is a subset of } S \text{ that sums to } B \}
\]

SUBSET-SUM is NP-complete

• We are reducing from the language:

3SAT = \{ \varphi : \varphi \text{ is a 3-CNF formula that has a satisfying assignment} \}

to the language:

\[
\text{SUBSET-SUM} = \{ (S = \{ a_1, a_2, a_3, \ldots, a_k \}, B) : \text{there is a subset of } S \text{ that sums to } B \}
\]

SUBSET-SUM is NP-complete

• \( \varphi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor x_3) \land \ldots \land (\ldots) \)

• Need integers to play the role of truth assignments

• For each variable \( x_i \) include two integers in our set \( S \):
  - \( x_i^{\text{TRUE}} \) and \( x_i^{\text{FALSE}} \)

• set \( B \) so that exactly one must be in sum

\[
\begin{align*}
x_1^{\text{TRUE}} &= 1000\ldots0 \\
x_1^{\text{FALSE}} &= 1000\ldots0 \\
x_2^{\text{TRUE}} &= 0100\ldots0 \\
x_2^{\text{FALSE}} &= 0100\ldots0 \\
\vdots \\
x_m^{\text{TRUE}} &= 0000\ldots1 \\
x_m^{\text{FALSE}} &= 0000\ldots1 \\
B &= 1111\ldots1
\end{align*}
\]

- every choice of one from each
  \((x_i^{\text{TRUE}}, x_i^{\text{FALSE}})\) pair sums to \( B \)
- every subset that sums to \( B \) must choose one from each \((x_i^{\text{TRUE}}, x_i^{\text{FALSE}})\) pair
SUBSET-SUM is NP-complete

- $\phi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor x_5) \land \ldots \land (\ldots)$
- Need to force subset to "choose" at least one true literal from each clause
- Idea:
  - add more digits
  - one digit for each clause
  - set $B$ to force each clause to be satisfied.

- $B = 1111 \ldots 1 ? ? ? ?$
- if clause $i$ is satisfied sum might be 1, 2, or 3 in corresponding column.
- want $? \geq 1$
- solution: set $? = 3$
- add two "filler" elements for each clause $i$:
  - $FILL_1 = 0000 \ldots 010\ldots 0$
  - $FILL_2 = 0000 \ldots 0010\ldots 0$
- column for clause $i$

Reduction: $m$ variables, $k$ clauses

- for each variable $x_i$:
  - $x_i^{\text{TRUE}}$ has ones in positions $k+i$ and $\{j : \text{clause } j \text{ includes literal } x_i\}$
  - $x_i^{\text{FALSE}}$ has ones in positions $k+i$ and $\{j : \text{clause } j \text{ includes literal } \neg x_i\}$
- for each clause $i$:
  - $FILL_1$ and $FILL_2$ have one in position $i$
  - bound $B$ has 3 in positions $1 \ldots k$ and $1$ in positions $k+1 \ldots k+m$

- $\text{NO maps to NO?}$
  - at most 5 ones in each column, so no carries to worry about
  - first $m$ digits of $B$ force subset to choose exactly one from each $(x_i^{\text{TRUE}}, x_i^{\text{FALSE}})$ pair
  - last $k$ digits of $B$ require at least one true literal per clause, since can only sum to 2 using filler elements
  - resulting assignment must satisfy $\phi$
MAX CUT

• Given graph $G = (V, E)$
  – a cut is a subset $S \subseteq V$
  – an edge $(x, y)$ crosses the cut if $x \in S$ and $y \notin S$ or $x \in V - S$ and $y \in S$
  – search problem: find cut maximizing number of edges crossing the cut

Theorem: the following language is NP-complete:
MAX CUT = \{$(G = (V, E), k)$ : there is a cut $S \subseteq V$ with at least $k$ edges crossing it\}

• Proof:
  – Part 1: MAX CUT $\in$ NP. Proof?
  – Part 2: MAX CUT is NP-hard.
    • reduce from?

Not-All-Equal 3SAT

$(x_1 \lor x_2 \lor \neg x_3)(\neg x_1 \lor x_4 \lor x_3) \land \ldots \land (\ldots)$

Theorem: the following language is NP-complete:
NAE3SAT = \{$\phi$ : $\phi$ is a 3-CNF formula for which there exists a truth assignment in which every clause has at least 1 true literal and at least 1 false literal\}

• Proof:
  – Part 1: NAE3SAT $\in$ NP. Proof?
  – Part 2: NAE3SAT is NP-hard. Reduce from?

NAE3SAT is NP-complete

• We are reducing from the language:
CIRCUIT-SAT = \{$C : C$ is a Boolean circuit for which there exists a satisfying truth assignment\}

to the language:
NAE3SAT = \{$\phi$ : $\phi$ is a 3-CNF formula for which there exists a truth assignment in which every clause has at least 1 true literal and at least 1 false literal\}
NAE3SAT is NP-complete

- Recall reduction to 3SAT
  - variables \(x_1, x_2, \ldots, x_n\), gates \(g_1, g_2, \ldots, g_m\)
  - produce clauses:
    - \(V\): output gate \(g_0\):
    - \(\lor (z_1 \vee z_2 \vee v w)\)
    - \(\lor (\neg z_1 \vee z_2 \vee v w)\)
    - \(\lor (\neg z_1 \vee \neg z_2 \vee v w)\)
    - \(\lor (\neg z_1 \vee z_2 \vee v w)\)
    - \(\lor (\neg z_1 \vee \neg z_2 \vee v w)\)
    - \(\lor (\neg z_1 \vee v w)\)

MAX CUT

- Given graph \(G = (V, E)\)
  - a cut is a subset \(S \subseteq V\)
  - an edge \((x, y)\) crosses the cut if \(x \in S\) and \(y \in V - S\) or \(x \in V - S\) and \(y \in S\)
  - search problem:
    - find cut maximizing number of edges crossing the cut

Theorem: the following language is NP-complete:

\[
\text{MAX CUT} = \{(G = (V, E), k) : \text{there is a cut } S \subseteq V \text{ with at least } k \text{ edges crossing it}\}
\]

- Proof:
  - Part 1: MAX CUT ∈ NP. Proof?
  - Part 2: MAX CUT is NP-hard.
  - reduce from?
MAX CUT is NP-complete

- We are reducing from the language:

  \[
  \text{NAE3SAT} = \{ \varphi : \varphi \text{ is a 3-CNF formula for which there exists a truth assignment in which every clause has at least 1 true literal and at least 1 false literal} \}
  \]

  to the language:

  \[
  \text{MAX CUT} = \{(G = (V, E), k) : \text{there is a cut } S \subseteq V \text{ with at least } k \text{ edges crossing it} \}
  \]

MAX CUT is NP-complete

- The reduction:

  - given instance of NAE3SAT (n nodes, m clauses):
    \[
    (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_4 \lor x_5) \land \ldots \land (\neg x_2 \lor x_3 \lor x_4)
    \]
  - produce graph \( G = (V, E) \) with node for each literal
    - triangle for each 3-clause
    - parallel edges for each 2-clause

MAX CUT is NP-complete

- YES maps to YES

  - take cut to be TRUE literals in a NAE truth assignment
  - contribution from clause gadgets: \( 2m \)
  - contribution from \( (x_i, \neg x_i) \) parallel edges: \( 3m \)
  - set \( k = 5m \)

MAX CUT is NP-complete

- NO maps to NO

  - Claim: if cut has \( x_i, \neg x_i \) on same side, then can move one to opposite side without decreasing # edges crossing cut
  - Proof:
    - contribution from \( (x_i, \neg x_i) \) parallel edges: \( 3m \)
    - contribution from clause gadgets must be \( 2m \)
    - conclude: there is a NAE assignment

Claim: if cut has \( x_i, \neg x_i \) on same side, then can move one to opposite side without decreasing # edges crossing cut

- Proof:
  - contribution from \( (x_i, \neg x_i) \) parallel edges: \( 3m \)
  - contribution from clause gadgets must be \( 2m \)
  - conclude: there is a NAE assignment
coNP

- Is NP closed under complement?

Can we transform this machine:

\[ x \in L \quad x \notin L \]

\[ q_{\text{accept}} \quad q_{\text{reject}} \]

\[ x \in L \quad x \notin L \]

\[ q_{\text{accept}} \quad q_{\text{reject}} \]

into a machine with this behavior?

31

coNP

- language \( L \) is in \( \text{coNP} \) iff its complement (co-\( L \)) is in \( \text{NP} \)

- it is believed that \( \text{NP} \neq \text{coNP} \)

- note: \( \text{P} = \text{NP} \) implies \( \text{NP} = \text{coNP} \)

- proving \( \text{NP} \neq \text{coNP} \) would prove \( \text{P} \neq \text{NP} \)

- another major open problem...

32

coNP

- canonical \( \text{coNP} \)-complete language:

\( \text{UNSAT} = \{ \varphi : \varphi \text{ is an unsatisfiable } 3-\text{CNF formula} \} \)

- proof?

33

coNP

- another example

\( \text{3-DNF-TAUTOLOGY} = \{ \varphi : \varphi \text{ is a } 3-\text{DNF formula and for all } x, \varphi(x) = 1 \} \)

- proof?

34