Message from BoC

I'd like to discuss concerns about academic integrity in this class.

The Board of Control has already received, and processed, several reports of cheating. I am worried that students may be under the false impression that those who are cheating and getting away with it, and I want to assure you that this is not the case. You may be able to observe them cheating, but one of the disadvantages of the confidentiality required by the BoC is that you do not see the result of it. Several instances have already been caught already, and if you have concerns, feel free to email boc@caltech.edu.

Additionally, if you are found to have violated the Honor Code in this class after this point, the fact that you have had this warning will be considered in your case. Please consider how your actions affect your professor, your classmates, and your personal integrity.

Thank you, Board of Control Leadership

CS21 Lecture 21

Outline

• NP-complete problems: Hamilton path and cycle, Traveling Salesperson Problem
• NP-complete problems: Subset Sum
• NP-complete problems: NAE-3-SAT, max cut

Hamilton Path

• Definition: given a directed graph G = (V, E), a Hamilton path in G is a directed path that touches every node exactly once.
• A language (decision problem):
  HAMPATH = {(G, s, t) : G has a Hamilton path from s to t}

HAMPATH is NP-complete

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

HAMPATH = {(G, s, t) : G has a Hamilton path from s to t}

*Proof:*
  - Part 1: HAMPATH ∈ NP. Proof?
  - Part 2: HAMPATH is NP-hard.
    - reduce from?

HAMPATH is NP-complete

• We are reducing from the language:
  3SAT = { φ : φ is a 3-CNF formula that has a satisfying assignment }

to the language:

HAMPATH = {(G, s, t) : G has a Hamilton path from s to t}
HAMPATH is NP-complete

- We want to construct a graph from $\phi$ with the following properties:
  - a satisfying assignment to $\phi$ translates into a Hamilton Path from $s$ to $t$
  - a Hamilton Path from $s$ to $t$ can be translated into a satisfying assignment for $\phi$
- We will build the graph up from pieces called gadgets that "simulate" the clauses and variables of $\phi$.

The variable gadget (one for each $x_i$):

- $x_i \text{ true:}$
- $x_i \text{ false:}$

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

- How to ensure that all $k$ clauses are satisfied?
- need to add nodes
  - can be visited in path if the clause is satisfied
  - if visited in path, implies clause is satisfied by the assignment given by path through variable gadget

One clause gadget for each of $k$ clauses:

- Clause gadget allows "detour" from "assignment path" for each true literal in clause

$\phi = (x_1 \lor x_2 \lor \neg x_3) \land \neg x_1 \lor x_4 \lor x_3 \land \ldots \land (\ldots)$
HAMPATH is NP-complete

\[ \phi = (x_1 \lor x_2 \lor \neg x_3 \lor x_4 \lor \neg x_5 \lor x_6) \]

- \( f(\phi) \) is this graph (edges to/from clause nodes not pictured)
- \( f \) poly-time computable?
- \# nodes = \( O(km) \)

YES maps to YES?
- try to translate path into satisfying assignment
- if path has “intended” form, OK.

NO maps to NO?
- path has “intended form” unless return from clause gadget to different variable gadget

What can go wrong?
- path has “intended form” unless return from clause gadget to different variable gadget

Case 1 (positive occurrence of \( v \) in clause):
- path must visit \( y \)
- must enter from \( x, z \), or \( c \)
- must exit to \( z \) (\( x \) is taken)
- \( x, c \) are taken, can’t happen

Case 2 (negative occurrence of \( v \) in clause):
- path must visit \( y \)
- must enter from \( x \) or \( z \)
- must exit to \( z \) (\( x \) is taken)
- \( x \) is taken, can’t happen
Undirected Hamilton Path

- HAMPATH refers to a directed graph.
- Is it easier on an undirected graph?

A language (decision problem):

\[
\text{UHAMPATH} = \{(G, s, t) : \text{undirected G has a Hamilton path from s to t}\}\]

UHAMPATH is NP-complete

Theorem: the following language is NP-complete:

\[
\text{UHAMPATH} = \{(G, s, t) : \text{undirected graph G has a Hamilton path from s to t}\}\]

Proof:
- Part 1: UHAMPATH ∈ NP. Proof?
- Part 2: UHAMPATH is NP-hard.
  - reduce from?

UHAMPATH is NP-complete

We are reducing from the language:

\[
\text{HAMPATH} = \{(G, s, t) : \text{directed graph G has a Hamilton path from s to t}\}\]

to the language:

\[
\text{UHAMPATH} = \{(G, s, t) : \text{undirected graph G has a Hamilton path from s to t}\}\]

The reduction:

\[
\text{G} \rightarrow \text{G}'
\]

- replace each node with three (except s, t)
  - \((u_{i1})_{\text{in}}\)
  - \((u_{i1})_{\text{mid}}\)
  - \((u_{i1})_{\text{out}}\)
  - \((u_{i2})_{\text{in}}\)
  - \((u_{i2})_{\text{mid}}\)
  - \((u_{i2})_{\text{out}}\)

UHAMPATH is NP-complete

No maps to No?

- Hamilton path in G:
  - \(s_{\text{out}}, v_1, v_2, v_3, v_4, v_5, v_6, ..., v_k-2, v_k-1, v_k, t_{\text{in}}\)
  - \(v_i = (u_{i1})_{\text{in}}\) for some \(i\) (only edges to in)
  - \(v_i = (u_{i1})_{\text{mid}}\) for some \(i\) (only way to enter mid)
  - \(v_i = (u_{i1})_{\text{out}}\) for some \(i\) (only way to exit mid)
  - \(v_i = (u_{i2})_{\text{out}}\) for some \(i\) (only edges to out)
  - Hamilton path in G:
    - \(s, u_{i1}, u_{i2}, u_{i3}, ..., u_{ik}, t\)
Undirected Hamilton Cycle

• Definition: given a undirected graph \( G = (V, E) \), a Hamilton cycle in \( G \) is a cycle in \( G \) that touches every node exactly once.
• Is finding one easier than finding a Hamilton path?
• A language (decision problem):
  \[ \text{UHAMCYCLE} = \{G : G \text{ has a Hamilton cycle}\} \]

UHAMCYCLE is NP-complete

**Theorem:** the following language is NP-complete:
\[ \text{UHAMCYCLE} = \{G : G \text{ has a Hamilton cycle}\} \]
• Proof:
  – Part 1: UHAMCYCLE ∈ NP. Proof?
  – Part 2: UHAMCYCLE is NP-hard.
    • reduce from?

Traveling Salesperson Problem

• Definition: given \( n \) cities \( v_1, v_2, ..., v_n \) and inter-city distances \( d_{ij} \), a TSP tour in \( G \) is a permutation \( \pi \) of \( \{1...n\} \). The tour's length is \( \sum_{i=1}^{n} d_{\pi(i)\pi(i+1)} \) (where \( n+1 \) means 1).
• A search problem: given the \( \{d_{ij}\} \), find the shortest TSP tour
• corresponding language (decision problem):
  \[ \text{TSP} = \{(\{d_{ij}\} : 1 \leq i \leq j \leq n, k) : \text{these cities have a TSP tour of length} \leq k \} \]

TSP is NP-complete

**Theorem:** the following language is NP-complete:
\[ \text{TSP} = \{(\{d_{ij}\} : 1 \leq i \leq j \leq n, k) : \text{these cities have a TSP tour of length} \leq k \} \]
• We are reducing from the language:
  \[ \text{UHAMCYCLE} = \{G : G \text{ has a Hamilton cycle}\} \]
  to the language:
  \[ \text{TSP} = \{(\{d_{ij}\} : 1 \leq i \leq j \leq n, k) : \text{these cities have a TSP tour of length} \leq k \} \]
TSP is NP-complete

- The reduction:
  - given $G = (V, E)$ with $n$ nodes
  produce:
  - $n$ cities corresponding to the $n$ nodes
  - $d_{uv} = 1$ if $(u, v) \in E$
  - $d_{uv} = 2$ if $(u, v) \notin E$
  - set $k = n$

YES maps to YES?
- if $G$ has a Hamilton cycle, then visiting cities in that order gives TSP tour of length $n$

NO maps to NO?
- if TSP tour of length $\leq n$, it must have length exactly $n$.
- all distances in tour are 1. Must be edges between every successive pair of cities in tour.