CSE 332: Data Structures and Parallelism

Spring 2022

Richard Anderson

Lecture 28: NP-Completeness II

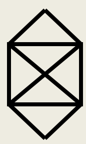
Announcements

- Final
  - Thursday, June 9, 8:30-10:20 AM, cse2 G20
  - No notes, calculator, internet, etc.
  - Comprehensive
    - All topics covered on the lecture slides
    - Estimate: 1/3rd pre-midterm, 2/3rd post-midterm
- Resources
  - Old exams
  - Review session, Tuesday, June 7, 3:00-4:30 PM, location TBD

Where we are in the story

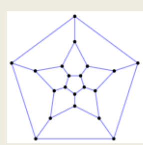
$e^{i\pi} = -1$

Eulerian Circuit: Easy, Polynomial time



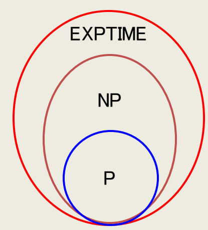
$i^2 + j^2 + k^2 = ijk = -1$

Hamiltonian Cycle: Seems hard, Exponential time



NP Completeness

- "Easy problems" – solvable in Polynomial Time
- Hard problems – take exponential time
- Interesting class of problems: Non-deterministic polynomial time

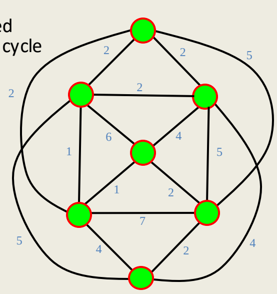


Travelling Salesman Problem

Given a connected, undirected graph with edge costs, find a cycle that includes all vertices with minimum total edge cost

Notes:

1) The cycle does not need to be simple, so it can visit vertices multiple times
2) The graph is often assumed to be complete. Missing edges can be given a cost of ∞.



How does TSP relate to HC?

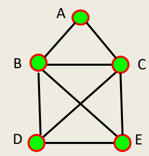
Problem Reduction

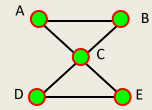
- Hamiltonian Circuit
  - Given a graph, is the a simple cycle that includes all of the vertices
- Travelling Salesman Problem
  - Given a complete graph with edge costs and a constant C, is there a cycle that visits all vertices with total cost at most C
- A reduction of HC to TSP uses an instance of TSP to solve an instance of HC
- This shows TSP is ***harder*** than HC

Transforming HC into TSP

- We can transform Hamiltonian Cycle into TSP.
- Given graph G=(V, E):
  - Assign weight of 1 to each edge
  - Augment the graph with edges until it is a complete graph G'=(V, E').
  - Assign weight of 2 to the new edges.
  - Let C = |V|.

Examples





What is NP?

- Problems solvable in non-deterministic polynomial time . . .

- Problems where "yes" instances have polynomial time checkable certificates

Certificate examples

- Independent set of size K
  - The Independent Set
- Satifisfiable formula
  - Truth assignment to the variables
- Hamiltonian Circuit Problem
  - A cycle including all of the vertices
- K-coloring a graph
  - Assignment of colors to the vertices

Certifiers and Certificates: 3-Satisfiability

SAT: Does a given CNF formula have a satisfying formula

Certificate: An assignment of truth values to the n boolean variables

Certifier: Check that each clause has at least one true literal

Instance

$$(\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3} \vee \overline{x_4})$$

Certificate

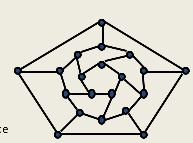
$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1$$

Certifiers and Certificates: Hamiltonian Cycle

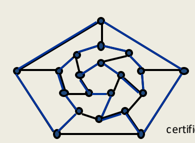
HAM-CYCLE. Given an undirected graph G = (V, E), does there exist a simple cycle C that visits every node?

Certificate. A permutation of the n nodes.

Certifier. Check that the permutation contains each node in V exactly once, and that there is an edge between each pair of adjacent nodes in the permutation.



instance



certificate

Polynomial time reductions

- Y is Polynomial Time Reducible to X
  - Solve problem Y with a polynomial number of computation steps and a polynomial number of calls to a black box that solves X
  - Notations: $Y <_P X$

Lemmas

- Suppose $Y <_P X$. If X can be solved in polynomial time, then Y can be solved in polynomial time.
- Suppose $Y <_P X$. If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time.

NP-Completeness

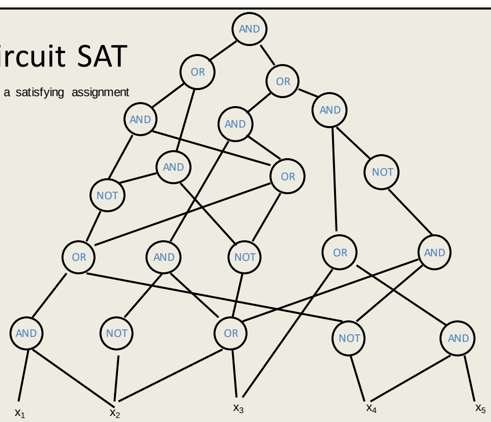
- A problem X is NP-complete if
  - X is in NP
  - For every Y in NP, $Y <_P X$
- X is a "hardest" problem in NP
- If X is NP-Complete, Z is in NP and $X <_P Z$
  - Then Z is NP-Complete

Cook's Theorem

- The Circuit Satisfiability Problem is NP-Complete

Circuit SAT

Find a satisfying assignment

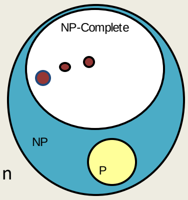


Proof of Cook's Theorem

- Reduce an arbitrary problem Y in NP to Circuit SAT
- Let A be a non-deterministic polynomial time algorithm for Y
- Convert A to a circuit, so that instance I of Y is a Yes instance iff and only if the circuit is satisfiable

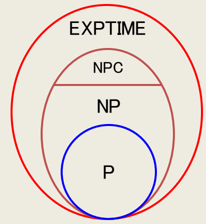
Populating the NP-Completeness Universe

- Circuit Sat $<_P$ 3-SAT
- 3-SAT $<_P$ Independent Set
- 3-SAT $<_P$ Vertex Cover
- Independent Set $<_P$ Clique
- 3-SAT $<_P$ Hamiltonian Circuit
- Hamiltonian Circuit $<_P$ Travelling Salesman
- 3-SAT $<_P$ Integer Linear Programming
- 3-SAT $<_P$ Graph Coloring
- 3-SAT $<_P$ Subset Sum
- Subset Sum $<_P$ Scheduling with Release times and deadlines



P, NP, NPC, and Exponential Time Problems

- All **currently known** algorithms for NP-complete problems run in **exponential** worst case time
- Diagram depicts relationship between P, NP, and EXPTIME (class of problems that **provably require** exponential time to solve)



It is believed that
$P \neq NP \neq EXPTIME$

Great Quick Reference

Is this lecture complete? Hardly, but here's a good reference:

Computers and Intractability: A Guide to the Theory of NP-Completeness
by Michael S. Garey and David S. Johnson