CS 170 Section 10

Search Problems and Intractability

Owen Jow

owenjow@berkeley.edu

4/04 Algorithm Not Found

Berkeley
UNIVERSITY OF CALIFORNIA

Search Problem

- Find a solution S to the problem instance I.
- A solution can be verified in polynomial time by the algorithm $C(I, S)$.

Examples

SAT: find a satisfying truth assignment for a Boolean formula.
TSP: find a tour [1] of total distance b or less.

[1] a cycle that passes through every vertex exactly once

Berkeley
UNIVERSITY OF CALIFORNIA

Optimization Problem

- Find the **best** solution *S* to the problem instance *I*.
- "Best" should be quantified by some objective function.

Examples

MAX-SAT: find the max number of clauses that can be simultaneously true.
TSP-OPT: find a tour of minimum distance.

Search vs Optimization

- Search and optimization formulations are of equal difficulty.
- Why? *Each reduces to the other.*

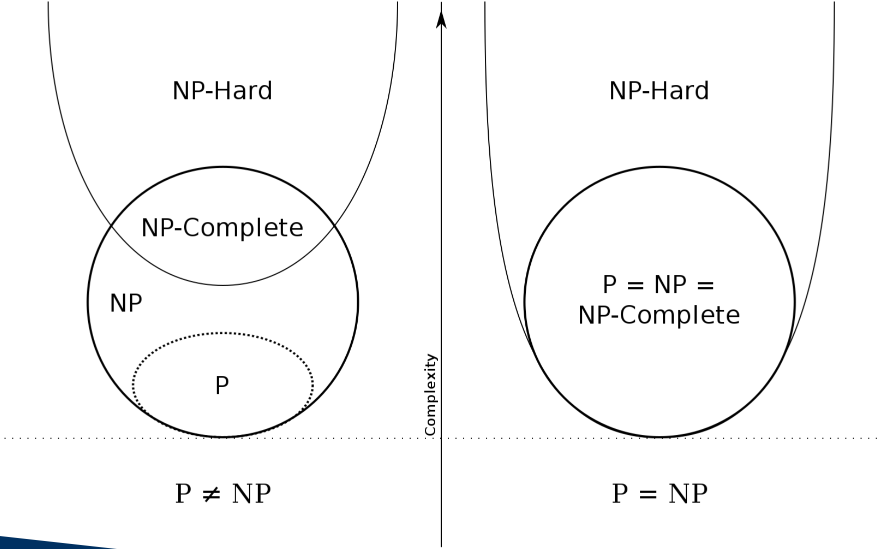
$$\textbf{TSP} \longleftrightarrow \textbf{TSP-OPT}$$

Berkeley
UNIVERSITY OF CALIFORNIA

P vs NP

- **P:** all search problems that can be solved in polynomial time
- **NP:** all search problems (i.e. "verifiable in polynomial time")
- **NP-complete:** the problems to which all search problems reduce
- **NP-hard:** "at least as hard as the NP-complete problems"

> I know an NP-complete joke, but once you've heard one you've heard them all.
>
> *jason*, Stack Overflow



Berkeley
UNIVERSITY OF CALIFORNIA

Examples

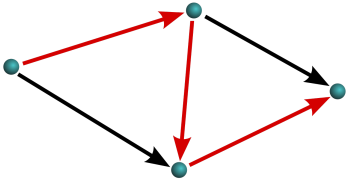
NP-complete	P
3SAT	HORN SAT
TSP	MST
ILP	LP
RUDRATA PATH	EULER PATH
BALANCED CUT	MINIMUM CUT
LONGEST PATH	SHORTEST PATH

Table 1: “Hard” versus “easy” search problems.

Berkeley
UNIVERSITY OF CALIFORNIA

A Faulty Reduction

- **Rudrata path:** find a path that goes through each vertex exactly once.
  - This is also known as the *Hamiltonian path* problem.
- **Longest path:** find a simple path of length $\geq g$ [search formulation].
  - *Simple*: cannot pass through any vertex more than once.



Berkeley
UNIVERSITY OF CALIFORNIA

A Faulty Reduction

Undirected `RUDRATA PATH` **can be reduced to** `LONGEST PATH` **in a DAG.**

Given a graph $G = (V, E)$, we can create a DAG as a directed DFS tree. If the longest path in this DAG has $|V| - 1$ edges, then there is a Rudrata path in G (since a simple path with $|V| - 1$ edges visits every vertex).

- What is wrong with the given justification for our reduction?

Berkeley
UNIVERSITY OF CALIFORNIA

A Faulty Reduction Solution

Undirected `RUDRATA PATH` can be reduced to `LONGEST PATH` in a DAG.

Given a graph $G = (V, E)$, we can create a DAG as a directed DFS tree. If the longest path in this DAG has $|V| - 1$ edges, then there is a Rudrata path in G (since a simple path with $|V| - 1$ edges visits every vertex).

- What is wrong with the given justification for our reduction?

  To fully justify a reduction, we need to prove that an original problem instance I has a solution **iff** reduced problem instance I' has a solution.

  – It is possible to produce a DAG without a length $|V| - 1$ path in cases where G *does* have a Rudrata path.

Berkeley
UNIVERSITY OF CALIFORNIA