Intractability II: P, NP, and NP-Complete

"Why did the computer get a ticket? It parked in the 'NP-Complete Solutions Only' zone!"

Arundathi Neelam

TA Lesson Creation Contest - INFO 6205 Program Structure and Algorithms

Introduction to Computational Complexity

Definition of computational complexity

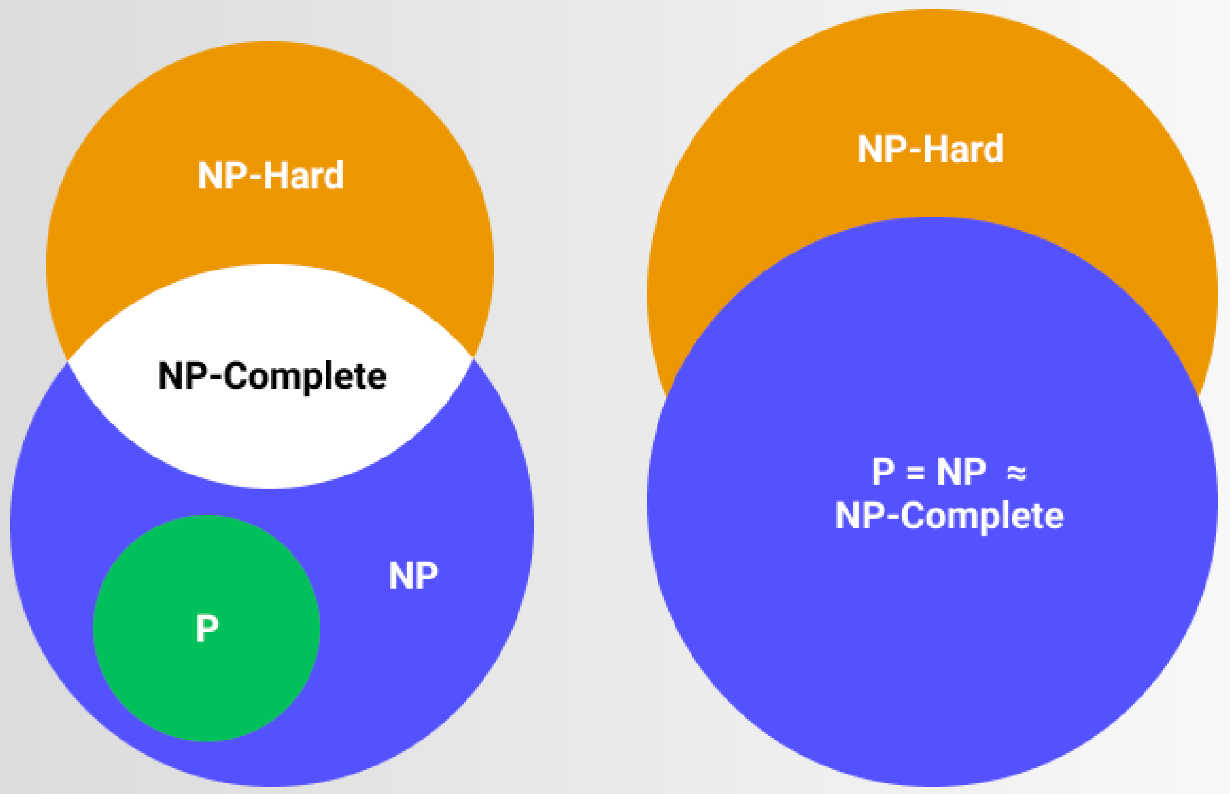
- Study of resources needed to solve problems.
- Measures time and space efficiency of algorithms.
- Classifies problems by difficulty and efficient solutions.

Importance of understanding algorithm efficiency

- Efficient algorithms save time and resources.
- Scalable for growing data sizes.
- Vital in fields like ML, cryptography, simulations.
- Enhances user experience and reduces energy consumption.

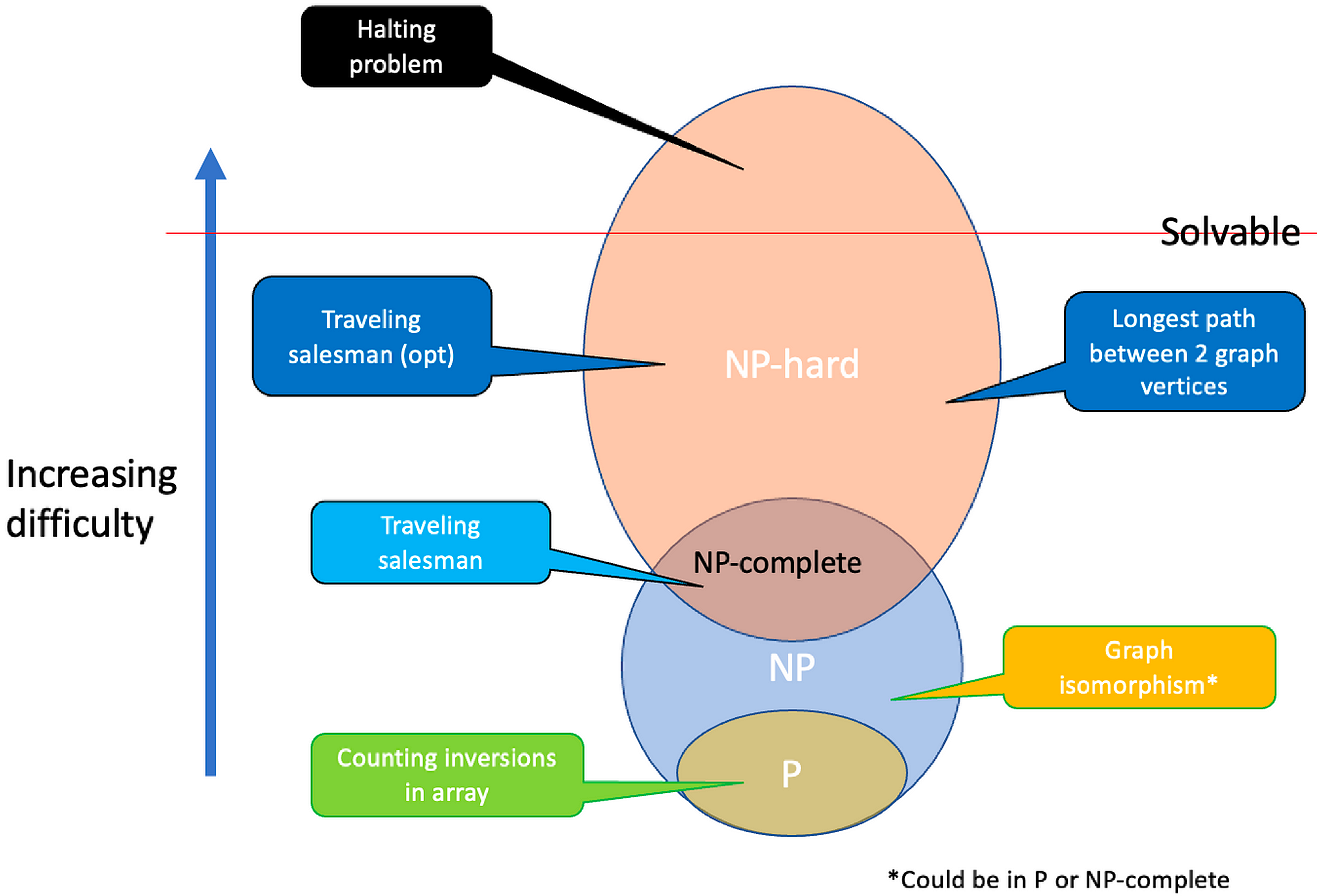
P, NP, and NP-Complete Classes

- P: Problems solvable fast with deterministic algorithms.
- NP: Problems verifiable quickly, solving might be hard.
- NP-Complete: Toughest NP problems, solving implies solving all.



P (Polynomial Time) Class

- P class includes problems solvable in reasonable time.
- Can be solved using algorithms with polynomial time complexity.
- Examples:
- Sorting, matrix multiplication, shortest path algorithms.



NP (Nondeterministic Polynomial Time) Class

- **Definition of the NP class:** The NP class includes decision problems for which a potential solution can be verified in polynomial time by a deterministic algorithm.

- **Problems verifiable in polynomial time:** NP problems are those for which if given a solution, it can be verified quickly, usually in polynomial time, by a deterministic algorithm.

- **Nondeterministic vs. deterministic algorithms:** Nondeterministic algorithms explore multiple paths simultaneously to guess a solution, while deterministic algorithms follow a single path to find a solution.

- **Examples of problems in NP:**

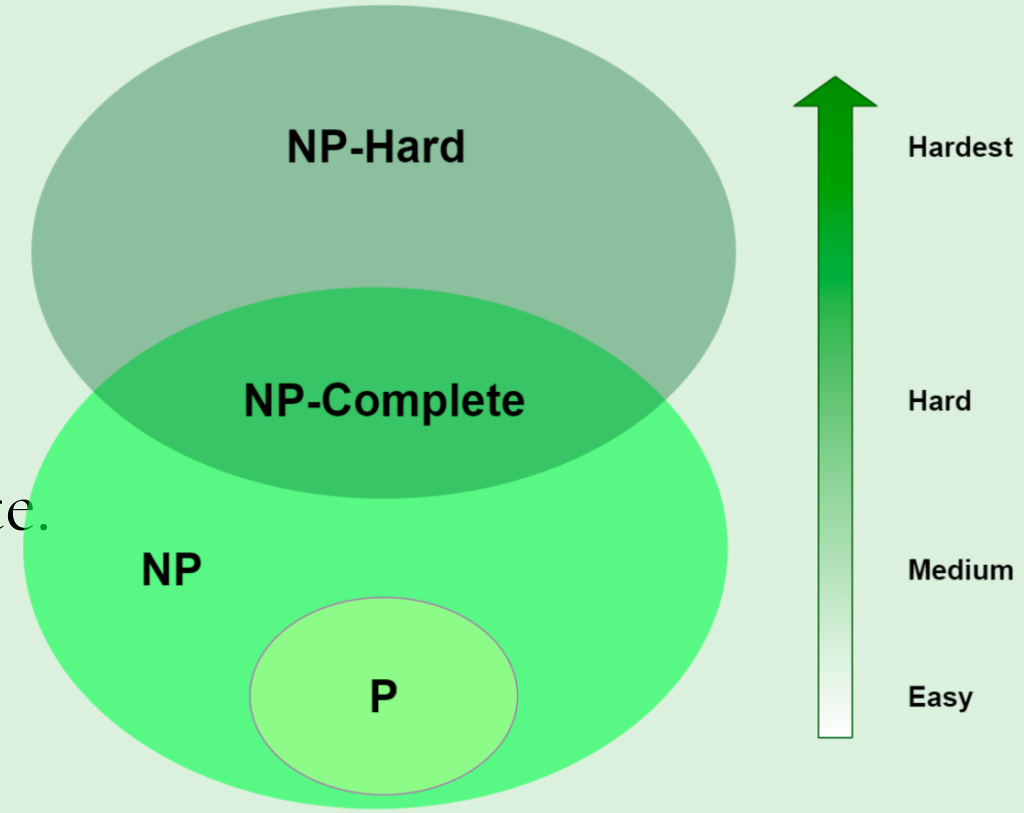
1. Traveling Salesman Problem: Given a list of cities and distances between them, is there a route that visits each city exactly once and returns to the starting city with a total distance less than a given value?
2. Boolean Satisfiability Problem (SAT): Given a Boolean formula, can we assign truth values to its variables to make the formula true?
3. Subset Sum Problem: Given a set of integers, is there a subset whose elements sum up to a given target value?
4. Graph Coloring Problem: Can we color the vertices of a graph such that no two adjacent vertices share the same color, using a specified number of colors?
5. Knapsack Problem: Given a set of items with weights and values, can we select a subset of items that maximizes the total value while staying within a given weight limit?

NP-Complete Class

- **Introduction to NP-Complete problems:** NP-Complete problems are a subset of NP problems that are believed to be among the most difficult and lack a known polynomial-time solution algorithm.

- **Definition of NP-Complete class:** NP-Complete problems are decision problems within the NP class that are at least as hard as the hardest problems in NP, meaning any NP problem can be transformed into an NP-Complete problem in polynomial time.

- **Cook's theorem and its significance:** Cook's theorem established the concept of NP-Completeness by showing that the Boolean Satisfiability Problem (SAT) is NP-Complete. This significance lies in demonstrating that many seemingly unrelated problems can be reduced to one another, helping classify their computational complexity.



- **Example of an NP-Complete problem:**

1. The Traveling Salesman Problem is an NP-Complete problem where the goal is to find the shortest possible route that visits a given set of cities and returns to the starting city, while visiting each city only once.

NP-Complete Reductions

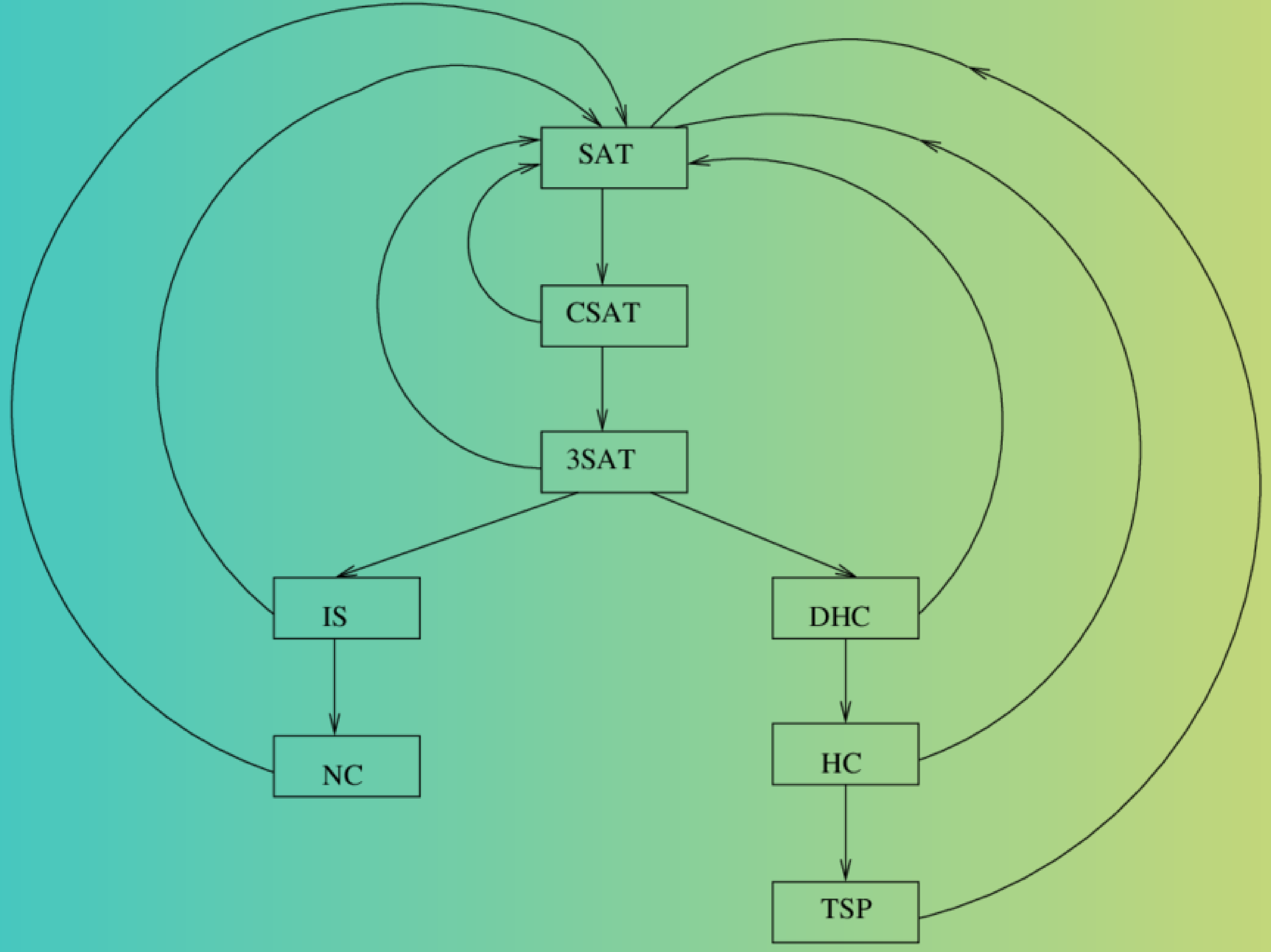
- **Explanation of problem reductions:** Problem reductions involve transforming one problem into another to show their computational equivalence or complexity relationships.

- **How NP-Complete problems are reduced to each other:** NP-Complete problems can be reduced to each other through polynomial-time transformations, demonstrating their shared complexity and the possibility of solving one if the other can be solved.

- **Diagrammatic representation of reductions:** Reductions are often depicted as directed arrows connecting problems, illustrating how solutions of one problem can be used to solve another, while maintaining their complexity relationships.

- **Illustrative example of a reduction:**

1. An example of an NP-Complete reduction is transforming the Subset Sum Problem into the Knapsack Problem, highlighting how the difficulty of solving one problem can be "reduced" to solving another.



SAT – Satisfiability Problem

CSAT – Satisfiability for formula in CNF Problem

3SAT – 3_Satisfiability Problem

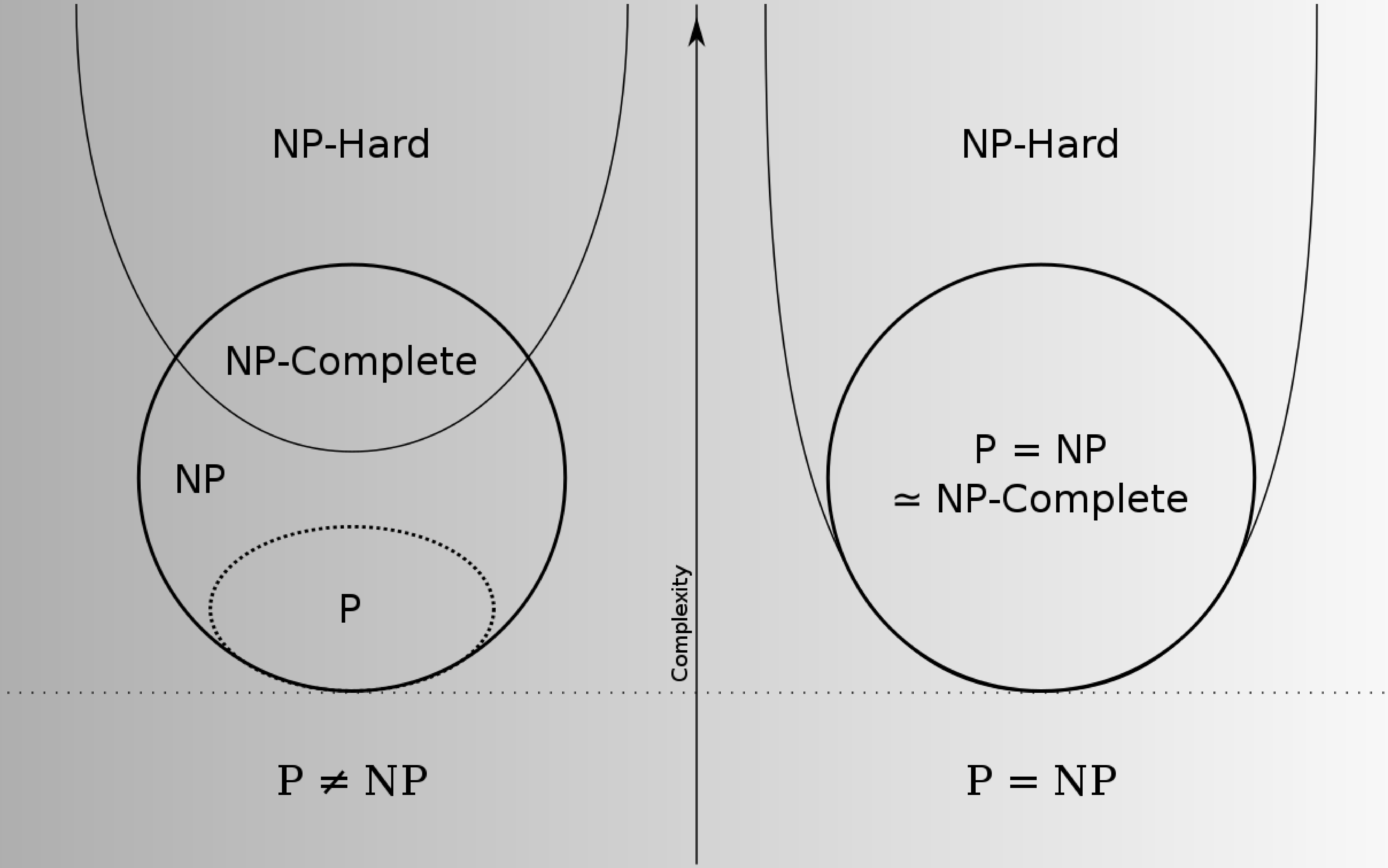
IS – Problem of Independent Sets

NC – Node−Cover Problem

DHC – Directed Hamilton−Circuit Problem

HC – Undirected Hamilton−Circuit Problem

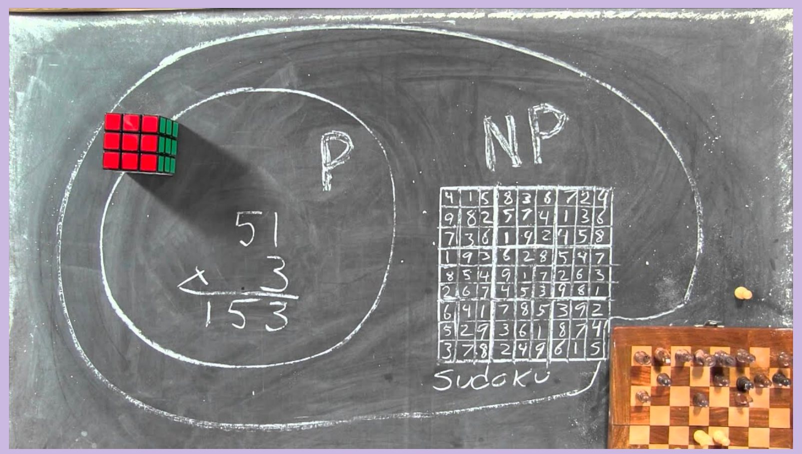
TSP – Traveling Salesman Problem


NP-Hard
NP-Complete
NP
P
P ≠ NP
Complexity
NP-Hard
P = NP
≃ NP-Complete
P = NP

Implications and Unsolved Questions

- **Importance of NP-Complete problems in computer science:** NP-Complete problems serve as benchmarks for computational difficulty, guiding algorithm design and revealing boundaries of efficient computation.

- **Theoretical implications of solving NP-Complete problems:** Solving an NP-Complete problem in polynomial time would imply P = NP, revolutionizing the field of algorithms and cryptography.



- **The P vs. NP problem and its open status:** The question of whether P (problems solvable in polynomial time) equals NP (problems verifiable in polynomial time) remains one of the most significant unsolved questions in computer science.

- Concluding remarks and encouraging further exploration: Recognizing the challenges posed by NP-Complete problems, we encourage ongoing research to tackle fundamental questions and advance our understanding of computation's limits.

Real-World Implications:

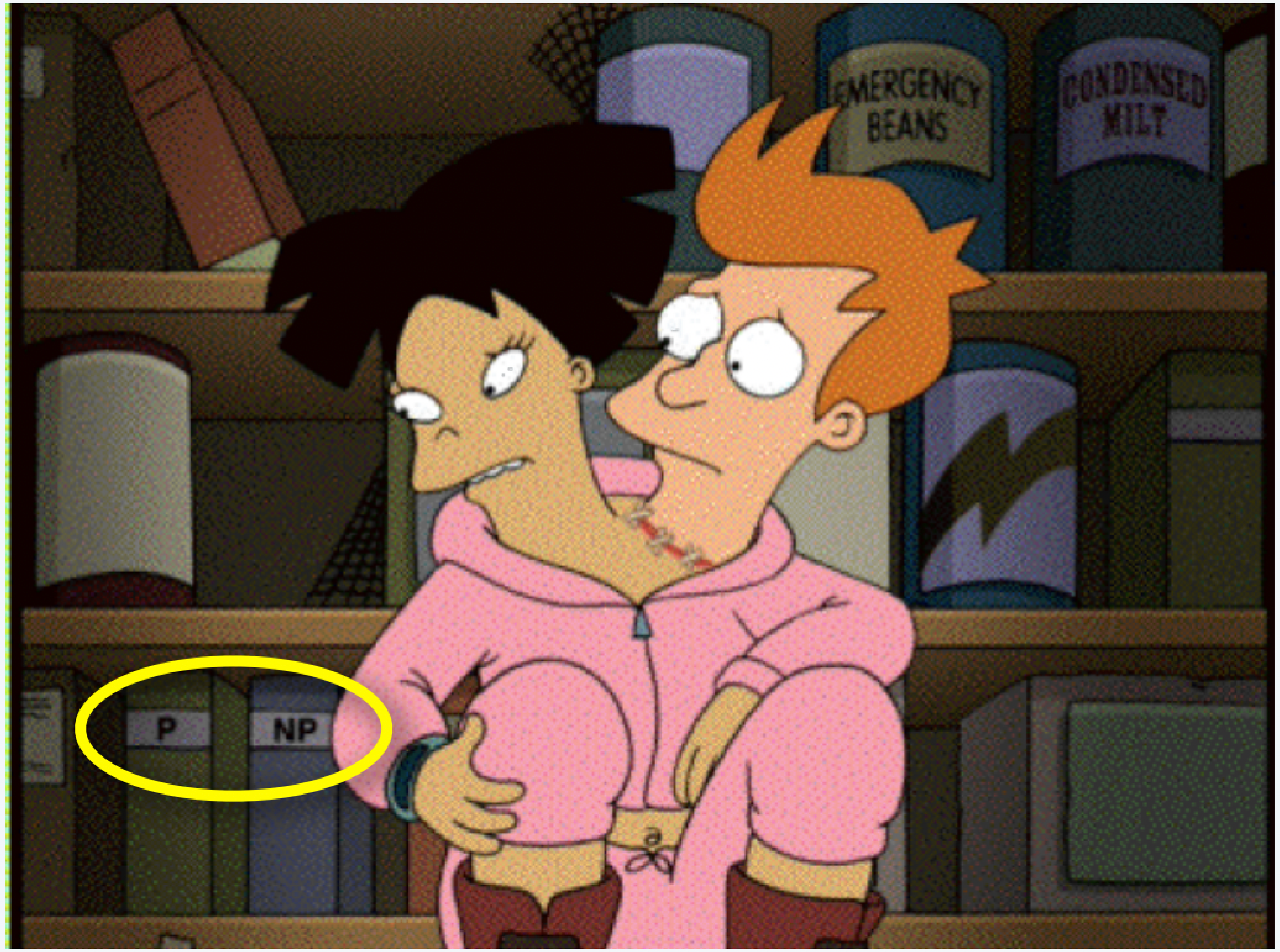
Circuit Design: The layout of integrated circuits involves solving optimization problems similar to NP-Complete problems. Designing efficient layouts is crucial for electronics manufacturing.

Network Routing: Optimizing data routing in computer networks involves solving problems related to shortest paths and connectivity, which are in the realm of P and NP.

DNA Sequence Alignment: Comparing DNA sequences to identify similarities and differences is a computational challenge that falls within the NP realm due to the complexity of sequence comparisons.

P vs. NP and pop culture

Copyright © 1990, Matt Groening



Copyright © 2000, Twentieth Century Fox

You NP-complete me

Thank you