MTech CSE – 1st Semester
Student ID: **A125023**
Student Name: **SRIJITA VERMA**

Question

Is the 2-SAT problem NP-hard? Can it be solved in polynomial time? Explain your reasoning.

Answer

Short Answer

The 2-SAT problem is not NP-hard (unless $\mathbf{P} = \mathbf{NP}$). Moreover, 2-SAT can be solved in polynomial time, and in fact in linear time, using graph-based algorithms such as implication graphs and strongly connected components (SCCs).

What Is the 2-SAT Problem?

The 2-SAT (2-CNF-SAT) problem is a restricted version of the Boolean satisfiability problem.

A Boolean formula is in *2-CNF* if:

- it is in conjunctive normal form (CNF), and
- each clause contains at most two literals.

For example:

$$(x_1 \vee \neg x_2) \ \wedge \ (\neg x_1 \vee x_3) \ \wedge \ (x_2 \vee x_3).$$

The problem asks whether there exists a truth assignment to the variables that satisfies all clauses.

Is 2-SAT NP-Hard?

Key Fact

2-SAT is not NP-hard unless $\mathbf{P} = \mathbf{NP}$.

Reasoning

The general SAT problem is NP-complete, and the restricted 3-SAT problem is also NP-complete. However, 2-SAT is a strictly easier special case.

If 2-SAT were NP-hard, then since 2-SAT can be solved in polynomial time, every problem in **NP** would also be solvable in polynomial time. This would imply:

$$\mathbf{P} = \mathbf{NP},$$

which is widely believed to be false. Hence, 2-SAT is not NP-hard.

Polynomial-Time Solvability of 2-SAT

Efficiency

2-SAT can be solved in:

$$O(n + m)$$

time, where:

- n is the number of variables,
- m is the number of clauses.

This efficiency is achieved using graph algorithms.

Implication Graph Formulation

Each 2-CNF clause

$$(a \vee b)$$

is logically equivalent to:

$$(\neg a \Rightarrow b) \ \wedge \ (\neg b \Rightarrow a).$$

Graph Construction

Construct a directed graph with:

- one node for each literal x and $\neg x$,
- for each clause $(a \vee b)$, add edges:

$$\neg a \to b \quad \text{and} \quad \neg b \to a.$$

This graph is called the *implication graph*.

Strongly Connected Components Criterion

Key Theorem

A 2-SAT formula is satisfiable if and only if, for no variable x, both x and $\neg x$ belong to the same strongly connected component of the implication graph.

Explanation

If $x \Rightarrow \neg x$ and $\neg x \Rightarrow x$, then assigning either truth value to x leads to a contradiction. Hence, no satisfying assignment exists.

Algorithm Outline

- Build the implication graph from the formula.
- Compute strongly connected components using Kosaraju's or Tarjan's algorithm.
- For each variable x:
  - if x and $\neg x$ lie in the same SCC, the formula is unsatisfiable;
  - otherwise, the formula is satisfiable and a valid assignment can be constructed.

All steps run in linear time.

Worked Example: Solving 2-SAT Using an Implication Graph

Consider the 2-CNF formula:

$$\varphi = (x_1 \vee x_2) \ \wedge \ (\neg x_1 \vee x_3) \ \wedge \ (\neg x_2 \vee \neg x_3).$$

We determine whether this formula is satisfiable using the implication graph method.

Conversion of Clauses into Implications

Each clause of the form $(a \vee b)$ is logically equivalent to:

$$(\neg a \Rightarrow b) \ \wedge \ (\neg b \Rightarrow a).$$

Applying this transformation to each clause:

- $(x_1 \vee x_2)$ yields:
$$\neg x_1 \Rightarrow x_2, \quad \neg x_2 \Rightarrow x_1.$$

- $(\neg x_1 \lor x_3)$ yields:
$$x_1 \Rightarrow x_3, \quad \neg x_3 \Rightarrow \neg x_1.$$
- $(\neg x_2 \lor \neg x_3)$ yields:
$$x_2 \Rightarrow \neg x_3, \quad x_3 \Rightarrow \neg x_2.$$

Construction of the Implication Graph

The vertices of the implication graph correspond to the literals:

$$\{ x_1, \neg x_1, x_2, \neg x_2, x_3, \neg x_3 \}.$$

The directed edges represent the implications derived above:

$$\neg x_1 \to x_2, \quad \neg x_2 \to x_1, \quad x_1 \to x_3, \quad \neg x_3 \to \neg x_1, \quad x_2 \to \neg x_3, \quad x_3 \to \neg x_2.$$

Analysis Using Strongly Connected Components

We now compute the strongly connected components (SCCs) of the implication graph.

For each variable x_i, we check whether the literals x_i and $\neg x_i$ belong to the same SCC. If they do, the formula is unsatisfiable.

In this implication graph:

- no variable and its negation appear in the same strongly connected component.

Conclusion for the Example

Since no variable x_i and its negation $\neg x_i$ lie in the same SCC, the formula φ is satisfiable.

A satisfying assignment can be constructed by assigning truth values according to the topological order of the SCC condensation graph.

Why 2-SAT Is Easier Than 3-SAT

The key difference lies in structure:

- 2-SAT constraints form implications that can be analyzed using graph reachability.
- 3-SAT constraints create complex combinatorial interactions that cannot be captured by simple implication graphs.

Thus, 2-SAT admits a global consistency check via SCCs, while 3-SAT remains NP-complete.

Complexity-Theoretic Classification

Problem	In P	NP-Hard	NP-Complete
SAT	No	Yes	Yes
3-SAT	No	Yes	Yes
2-SAT	Yes	No	No

Final Conclusion

2-SAT is not NP-hard (unless $\mathbf{P} = \mathbf{NP}$). It can be solved in polynomial time, and in fact in linear time, using implication graphs and strongly connected components.

This makes 2-SAT a classic example of how restricting problem structure can dramatically reduce computational complexity.

Intuition

2-SAT constraints behave like logical implications that must all be mutually consistent. Graph reachability is sufficient to detect contradictions. In contrast, 3-SAT allows richer interactions that require exponential search in the worst case.