Practical Quantum Computing: computational complexity and physics.

Jakub Tworzydło

Chair of Condensed Matter Physics
Institute of Theoretical Physicis

3/03/2026 Pasteura, Warszawa

Plan

Plan

Plan

Plan

Computational Complexity: A Physical Limit

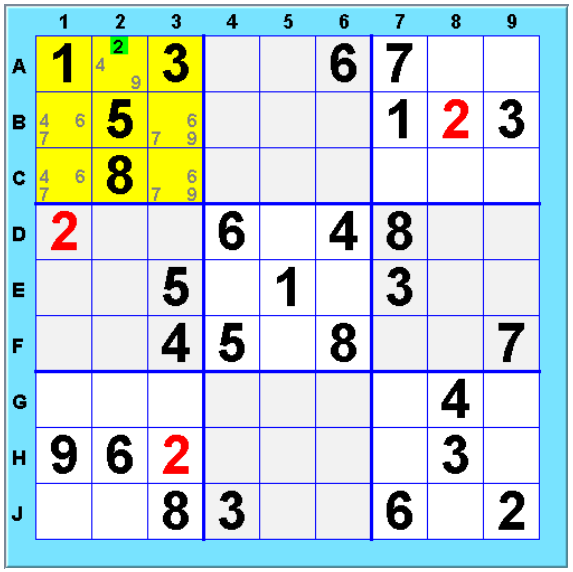
What is computational complexity?
The study of how resources (CPU time, memory) scale with the size N of the problem input.

Polynomial vs. Exponential Scaling

- Polynomial ($t \sim N^2, N^3$): Tractable. As computers get faster, we can solve significantly larger problems.
- Exponential ($t \sim 2^N$): Intractable. Buying a computer that is twice as fast only allows you to increase N by 1!

Many problems in physics and computer science involve finding a specific answer among an exponentially large space of possibilities.

The ”Needle in a Haystack”: Sudoku



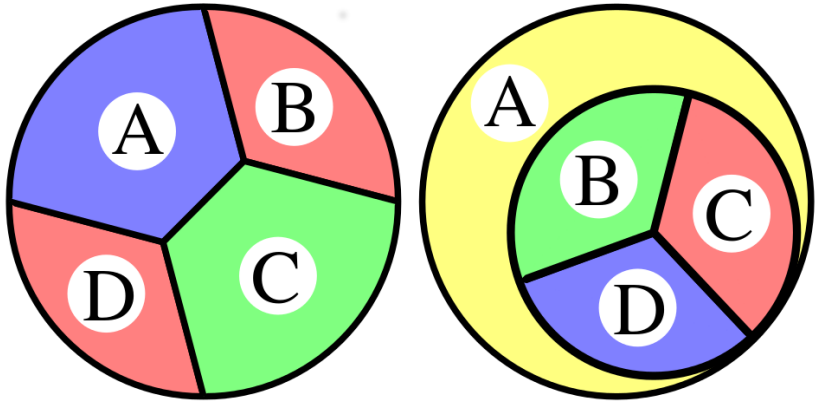
Consider an $N \times N$ Sudoku grid.

- **Finding the solution:** As N grows, the number of possible number combinations explodes exponentially. It is incredibly hard to find the solution from scratch.
- **Verifying the solution:** If someone hands you a completed grid, checking if it is correct takes only seconds (just scan rows, columns, and subgrids).

Optimization Example: Graph Coloring

The Problem:
Can the N nodes of a graph be colored with 3 colors (e.g., Red, Green, Blue) so that no two nodes joined by an edge share the same color?

- Like Sudoku, verifying a colored graph is trivial (just check each edge).
- Finding the valid coloring for a highly connected graph requires searching an exponentially large configuration space (3^N).

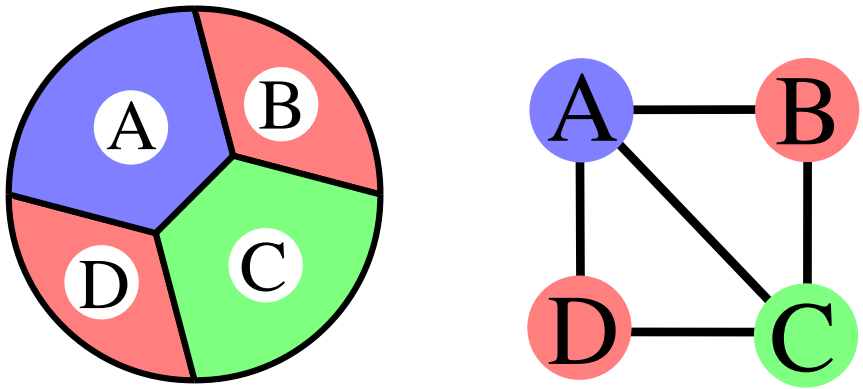


Note: The famous Map Coloring problem is just a special case formulated for planar graphs!

Optimization Example: Graph Coloring

The Problem:
Can the N nodes of a graph be colored with 3 colors (e.g., Red, Green, Blue) so that no two nodes joined by an edge share the same color?

- Like Sudoku, verifying a colored graph is trivial (just check each edge).
- Finding the valid coloring for a highly connected graph requires searching an exponentially large configuration space (3^N).



Note: The famous Map Coloring problem is just a special case formulated for planar graphs!

The Core Example: Boolean Satisfiability (SAT)

Suppose we have a long logical expression involving N boolean variables ($x_i \in \{\text{True, False}\}$).

Operations: NOT ($\neg$), AND ($\wedge$), OR ($\vee$)

CNF

means

Conjunctive Normal Form

by allacronyms.com

The Core Example: Boolean Satisfiability (SAT)

k-SAT Problem: The formula (CNF) is grouped into ”clauses” connected by ANDs. Each clause contains exactly k variables connected by ORs. There are M clauses in the logical expression.

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_4 \vee \neg x_5) \wedge \ldots$$

Question: Is there an assignment of True/False to the N variables that satisfies the entire expression?

Example Mapping: Coloring to 3SAT

We can rewrite 3-coloring problem in a SAT form:

A_R, A_G, A_B – variables for node A in red, blue and green

1) $A_R \vee A_G \vee A_B$ – condition that one of the colors is used
2) $\neg A_R \vee \neg A_G$ – the color is not red and green simultanously
3) $(\neg A_R \vee \neg B_R) \wedge (\neg A_G \vee \neg B_G) \wedge (\neg A_B \vee \neg B_B)$ – A and B vertices are not of the same color

Plan

Complexity Classes: P and NP

We can categorize these ”needle in a haystack” problems:

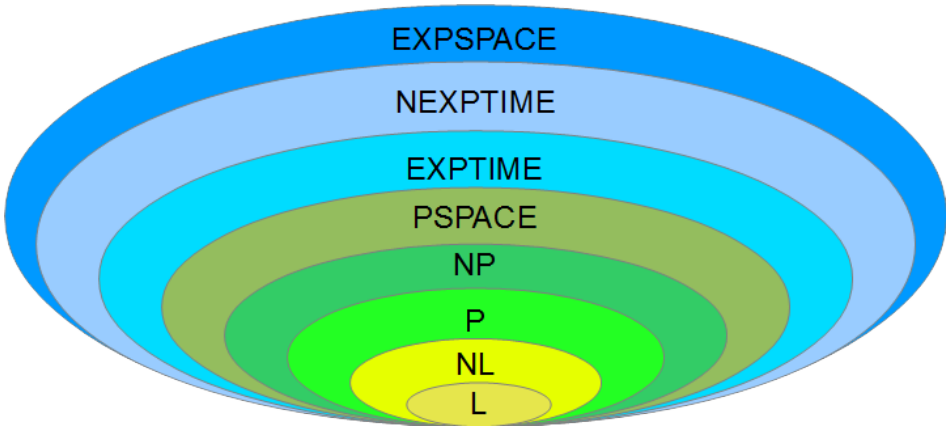
Class P (Polynomial Time):
Problems that can be *solved* fast (in polynomial time).

- Example: Sorting a list, shortest path on a map, and **2-SAT**.

Class NP (Non-deterministic Polynomial Time):
Problems where a proposed solution can be *verified* fast (in polynomial time).

- Example: Sudoku, Graph Coloring, and **3-SAT**.

The Map of Complexity



Complexity Class	Description
L, LSPACE	Logarithmic space, deterministic
NL	Logarithmic space, non-deterministic
P, PTIME	Polynomial time, deterministic
NP	Polynomial time, non-deterministic
PSPACE	Polynomial space
EXP, EXPTIME	Exponential time, deterministic
NEXP, NEXPTIME	Exponential time, non-deterministic
EXPSPACE	Exponential space

- Every problem in P is automatically in NP (if you can solve it fast, you can verify it fast).
- The biggest open question in computer scienc: Does $P = NP$? (If we can verify it fast, is there a hidden way to solve it fast?)

Universality: NP-Completeness

Computer scientists discovered that problems in NP can be **reduced** (translated) into one another.

- A Sudoku grid can be written as a list of logical SAT clauses.
- Graph coloring constraints are just logical constraints.

NP-Complete:
3-SAT is NP-Complete. It is a ”universal” problem. If we can build an algorithm or a physical machine that solves 3-SAT efficiently, we have instantly solved Sudoku, Graph Coloring, and *every* other problem in NP!

Plan

The Physics Dictionary: Spin Glasses

We can translate *NP*-complete optimization problems directly into the language of statistical mechanics using the **Ising Model**.

$$E(s_1, \ldots, s_N) = -\sum_{ij} J_{ij} s_i s_j - \sum_i h_i s_i$$

Boolean Variables (True/False) $\longleftrightarrow$ Ising Spins $s_i \in \{+1, -1\}$

Logical Constraints (Clauses) $\longleftrightarrow$ Couplings J_{ij} and fields h_i

Finding the SAT Solution $\longleftrightarrow$ Finding the Ground State Energy

Frustration and Complexity:
When couplings J_{ij} have mixed signs, spins cannot satisfy all interactions simultaneously (frustration). This creates a **Spin Glass** with a highly rugged energy landscape.

- 2D Planar Spin Glass $\rightarrow$ Polynomial Time (*P*)
- 3D or Non-Planar Spin Glass $\rightarrow$ ***NP*-Complete**

Explicit Mapping: SAT to an Ising Hamiltonian

How do we build the Hamiltonian? We assign an **energy penalty** ($E > 0$) only when a clause is violated.

Consider a clause: $(x_1 \vee \neg x_2)$. This is False *only* if $x_1 = \text{False}$ ($s_1 = -1$) and $x_2 = \text{True}$ ($s_2 = +1$). The energy penalty for this specific clause is:

$$H_{\text{clause}} = \frac{1}{4}(1 - s_1)(1 + s_2)$$

Note that $H_{\text{clause}} = 1$ if violated, and 0 if satisfied.

The total Hamiltonian is the sum over all M clauses:

$$H_{\text{total}} = \sum_{m=1}^{M} H_m(s) \quad \Longrightarrow \quad \text{Ground State } E = 0 \text{ means Satisfiable!}$$

Hint for students: For a comprehensive catalog of mapping NP problems to Ising models, see Andrew Lucas, "Ising formulations of many NP problems" (Front. Phys., 2014).

ŚolvingŃP-Complete Problems with Physics

If finding the solution is mathematically equivalent to finding the ground state of a spin glass, how does nature do it?

Simulated Annealing:

- Introduce a fictitious “temperature” T.
- Start at high T: thermal fluctuations allow the system to explore the entire rugged energy landscape, hopping over barriers.
- Slowly lower $T \to 0$: the system eventually freezes into the global minimum (the solution).

In 2002, Mézard, Parisi (Nobel 2021), and Zecchina applid the statistical mechanics methods to computer science.

Let Nature Compute: The Quantum Pivot

The limitation of classical annealing:
It relies on thermal energy to hop *over* high energy barriers. In a highly frustrated spin glass, these barriers can be impossibly high.

The Quantum Solution:
What if, instead of hopping over the barrier, we use quantum fluctuations to **tunnel through** it?

Next Week: Adiabatic Quantum Computing (AQC)

Suggested (short) presentations

- time scales, size scales for different quantum computing platforms
- algorithms: DPLL for kSAT, blossom al. for spin glass or similar
- solving classical spin glass with D-wave packages
- summary of today's lab