Quantum Computing and Quantum Machine Learning Hackathon

Use case 5:

Solving the k-SAT problem

with different techniques from quantum optimization

Juan Santana, Leendert van Egmond, Jakob Murauer, Younes Naceur

Overview

1. **Introduction:** k-SAT and ESG problems
2. **Methods:** Grover, QAOA, Quantum Annealing
3. **Random SAT-problems,** benchmarking of the methods
4. Accuracy analysis **with noise**
5. **Exploring** the feasible solutions
6. **Outlook:** UNSAT and LLM’s

k-SAT and ESG problems

ESG

- Environmental, **S**ocial, **G**overnance
- **Key factors** shaping investment decisions
- **Examples:** Inclusion, clean water, etc.
- **Factors to be fulfilled** (good governance) vs. **Factors to be avoided** (weapon export)

k-SAT

- **Satisfiability problem:** Given **m** logical clauses on **k** binary instances, is it possible to fulfil them?
- **Example:** find

$$x = (x_1, x_2) \text{ with } x_i \in \{0,1\}$$

s.t. $f(x) = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2)$

is **true**

Our Task:

- **Context:** Investor trying to invest in a portfolio of companies respecting his ESG demands
- **Input:** A set of companies, and their attributes (desirable + undesirable)
- **Objective:** Find portfolio of companies respecting a given set of values
- **Methods:** Quantum Approximate Algorithm, Grover’s algorithm, Quantum annealing

Grover vs. QAOA vs. Quantum Annealing

Three methods to solve the k-SAT problem

Grover's algorithm

- **Search algorithm with expected quadratic speedup**
- **Search in the space of possible solutions (binary strings) until one is feasible**
- **Implemented using the CLASSIQ library**

QAOA

- **Variational quantum algorithm:** Encrypt solution in ground state of Hamiltonian, compute ground state with **Variational Ansatz**
- **PYOMO:** Python library to implement combinatorial optimization problems
- Problems solved using **CLASSIQ**

Quantum Annealing

- **Analog quantum computing Ansatz:** Encrypt solution in ground state Hamiltonian
- **Shift Hamiltonian** from *easy* Hamiltonian to problem Hamiltonian
- Perform **adiabatic** time evolution to stay in **ground state**

Benchmarking with random SAT-problems

Way to benchmark our optimisers

- **Observation:** Can encrypt any k-SAT problem into a **k x m** matrix!
- **Translate matrix** as input to different solvers, compare **performance** on **random problems**
- **Issue:** Matrices have to encrypt **feasible** problems

$$C_1 = x_1 \qquad C_2 = x_1 \vee x_3$$

$$C_3 = \neg x_2 \qquad C_4 = \neg x_3$$

$$\xleftrightarrow{\textbf{1:1}}$$

$$\begin{array}{cc} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} & \begin{array}{c} C_1 \\ C_2 \\ C_3 \\ C_4 \end{array} \end{array}$$

Careful: Check that random matrices encrypt solvable problems!

Benchmarking with random SAT-problems

Way to benchmark our optimisers

QUANTUM ANNEALER

RANDOM k-SAT MATRIX → QAOA SOLVER → SOLUTION TO RANDOM ESG PROBLEM

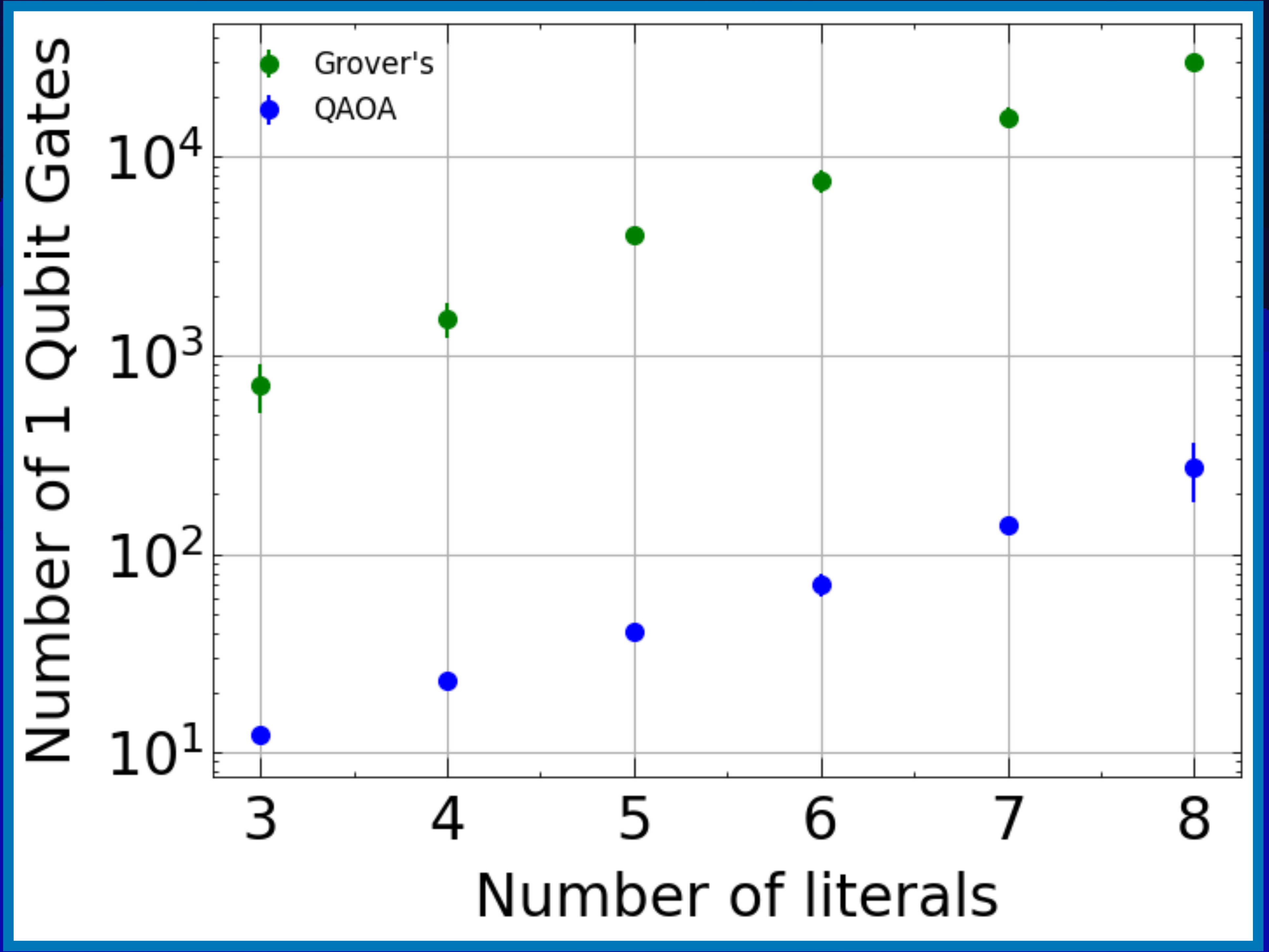
GROVER SOLVER

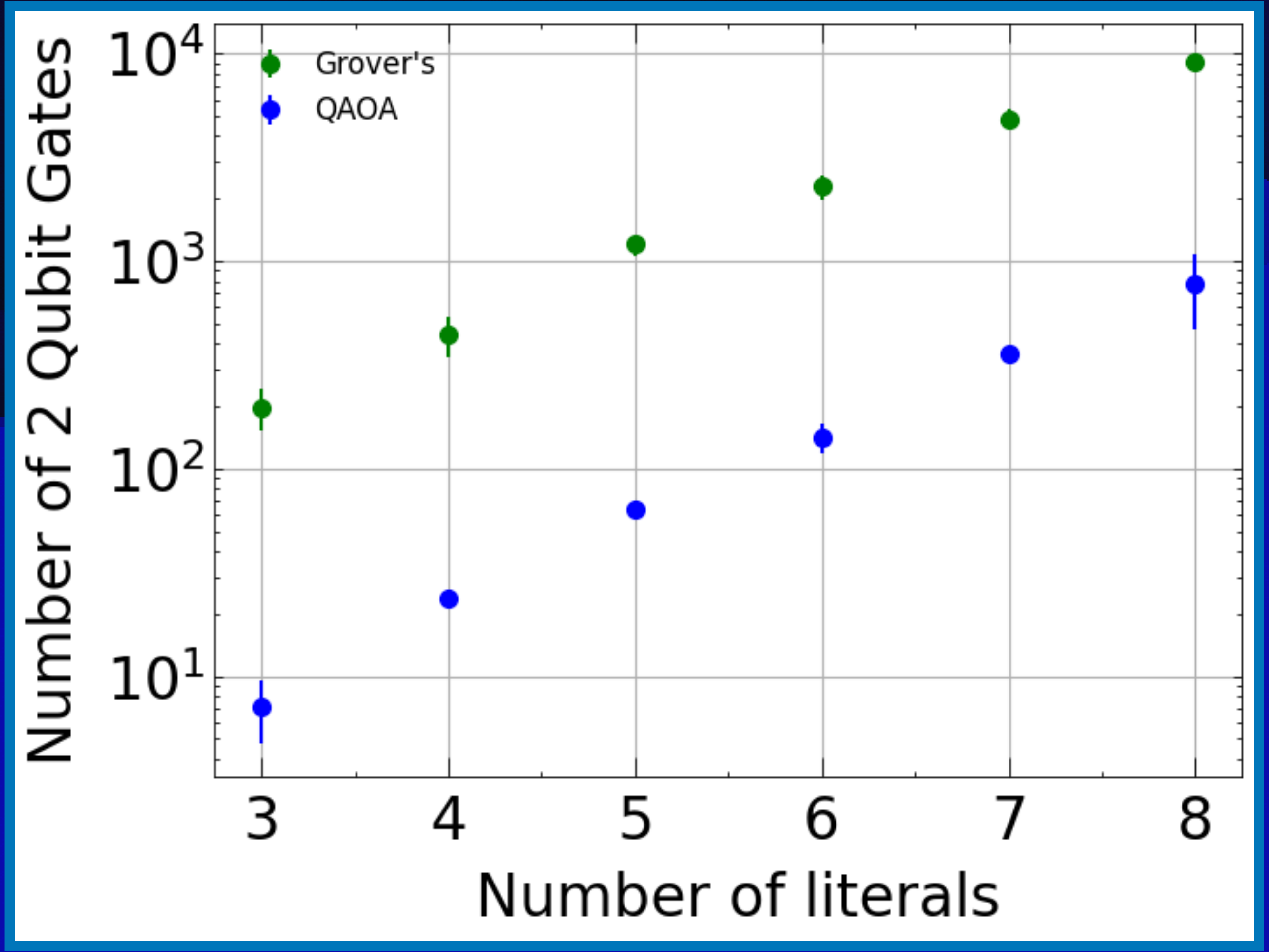
Remark: Included condition $x_1 \vee x_2 \ldots \vee x_k$ **to exclude trivial solutions** (no companies to invest in)

Number of gates needed per approach

k = # Literals

M = # Clauses





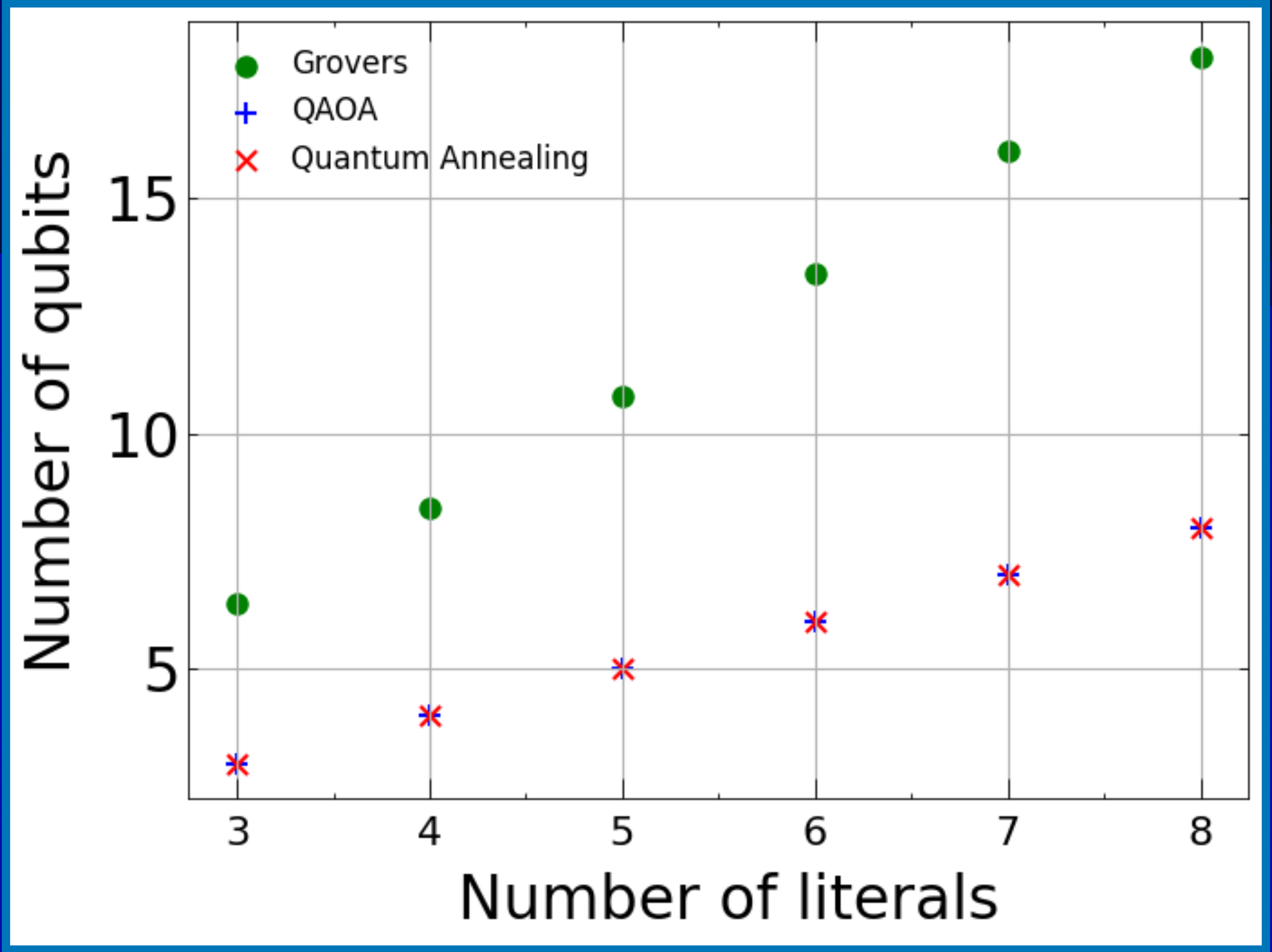
$k = M$

Quantum Annealing not included, doesn't use gates!

Number of qubits N needed per approach

k = # Literals

M = # Clauses



$k = M$

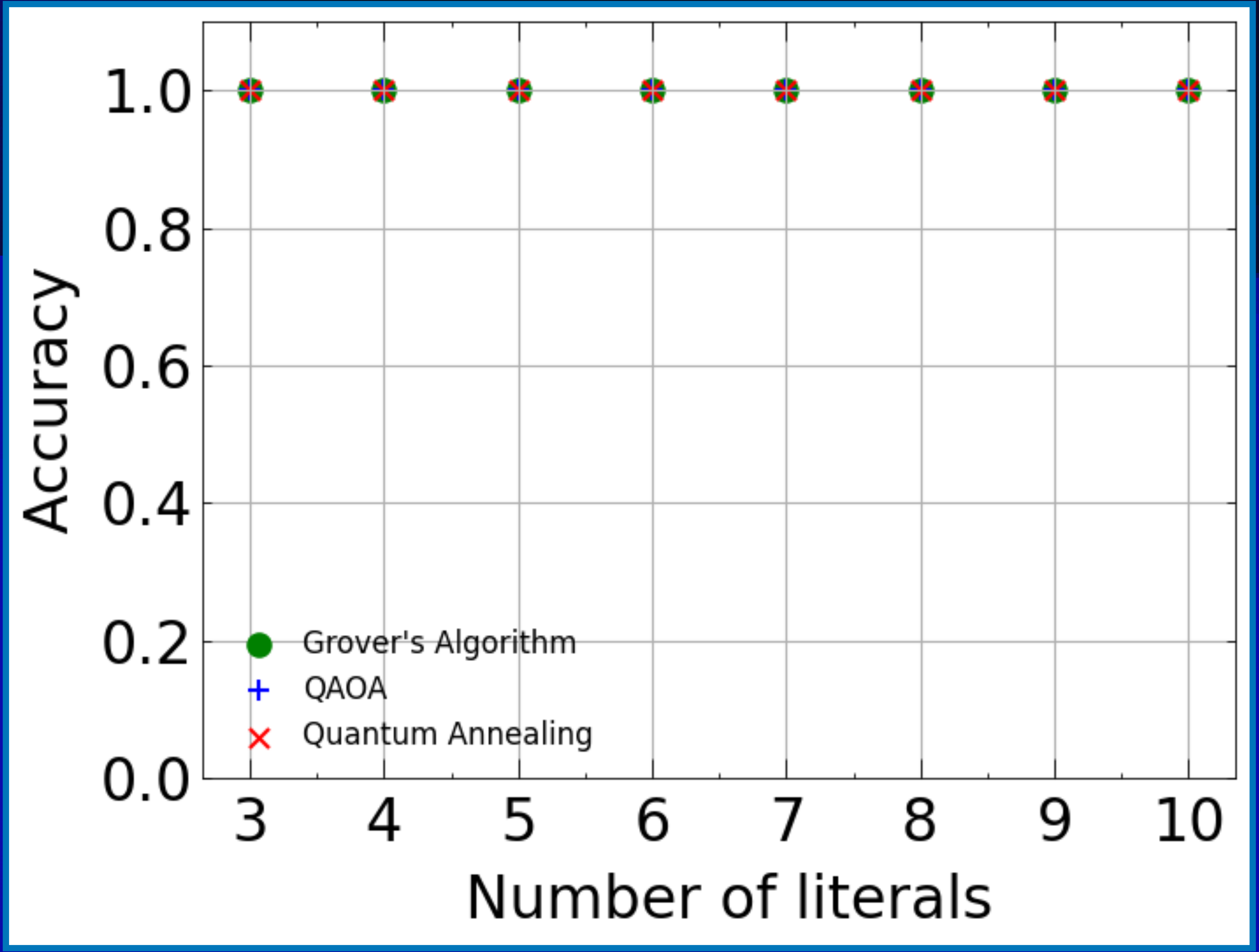
→ Quantum annealing and QAOA: $N = k$

→ Grover's algorithm: Linear scaling of N

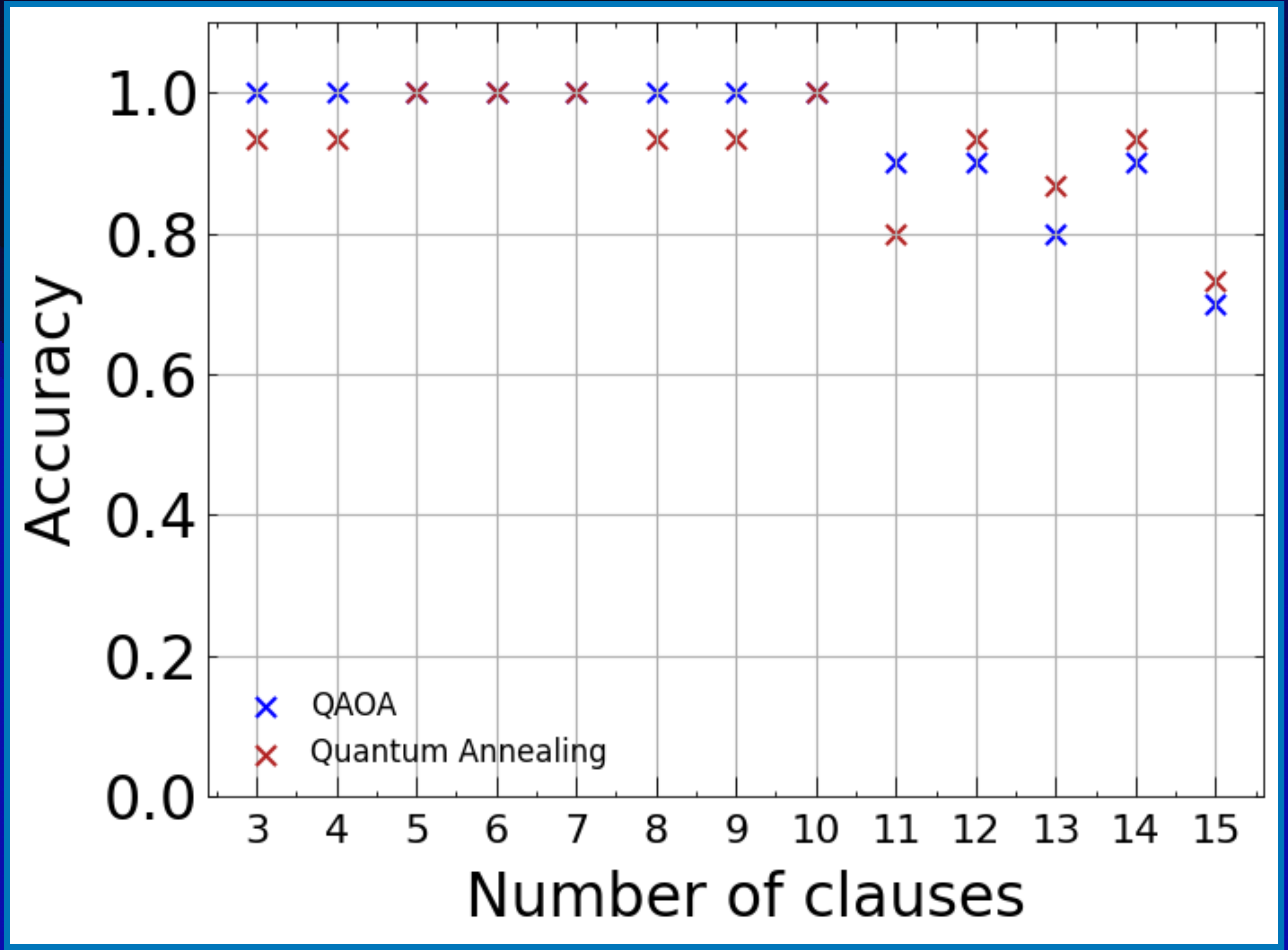
Accuracy analysis without noise

k = # Literals

M = # Clauses

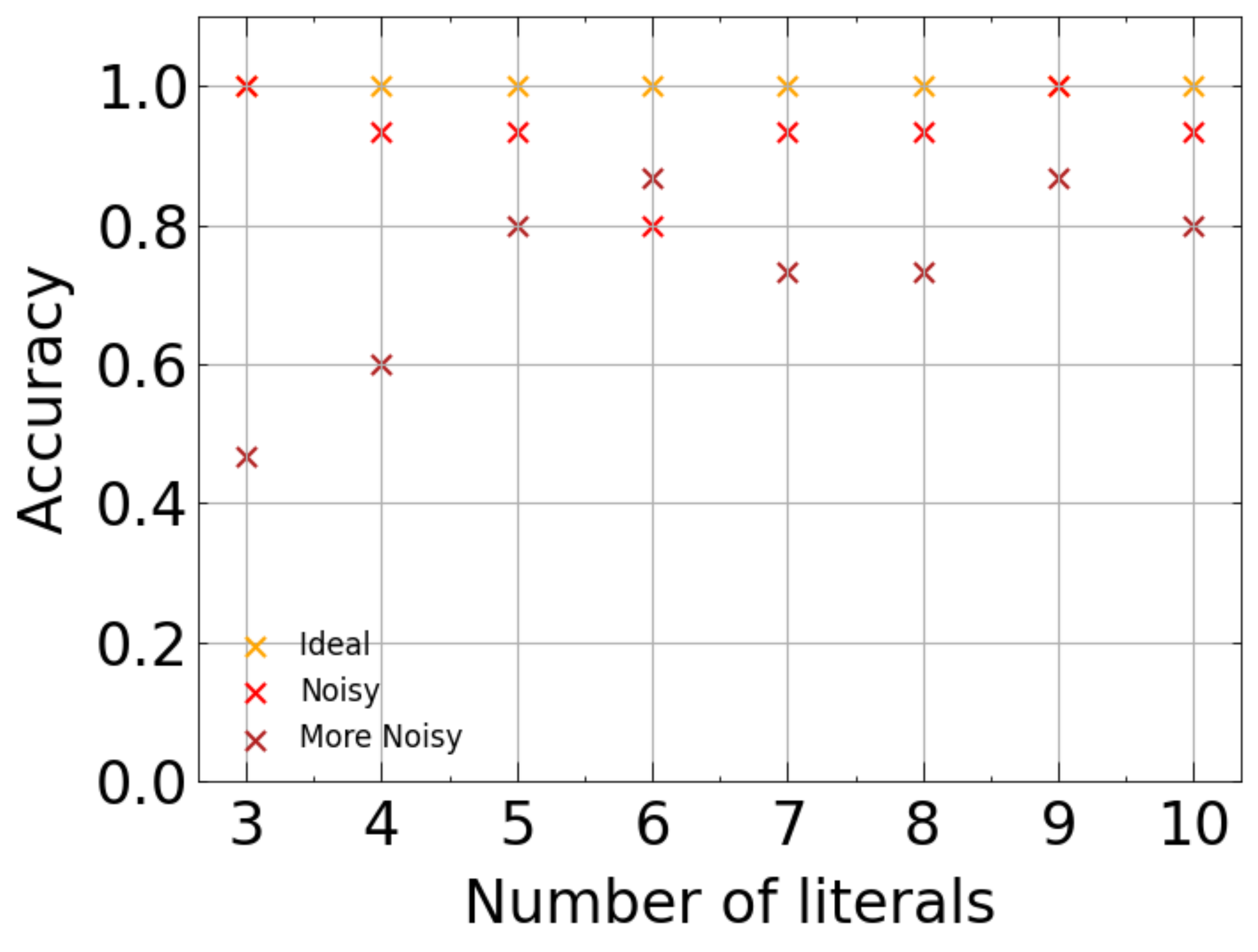


$k = M$



$k = 6,\ M$ variable

Accuracy analysis with noise for Quantum Annealing



$k = M$

k = # Literals

M = # Clauses

Exploring the feasible solutions

- **Observation:** For many k-SAT problems **several solutions are possible**
- **Idealised model:** As soon as a set of companies fulfills ESG conditions, **we invest!**
- **Not quite realistic!** ⟶ Compare the feasible solutions with other criteria
- **More adequate model:** Include portfolio optimization in space of feasible solutions

---

ALL SOLUTIONS —K-SAT SOLVING→ **FEASIBLE SOLUTIONS** —PORTFOLIO OPTIMIZATION→ **BEST SOLUTION**

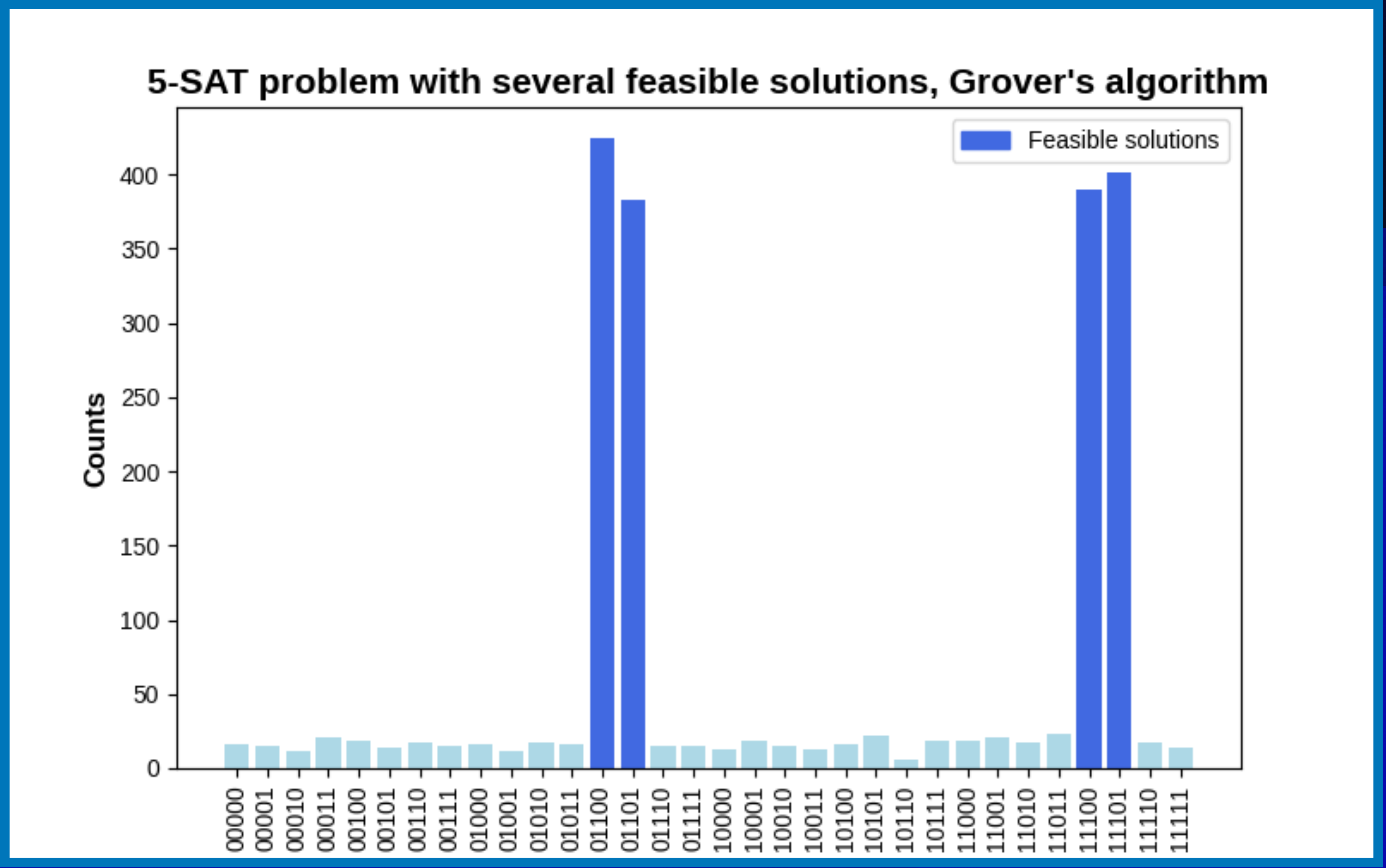
ALL SOLUTIONS
- All possible combinations of companies

FEASIBLE SOLUTIONS
- Companies omitting the ESG conditions

BEST SOLUTION
- Companies omitting the ESG conditions & maximising expected return

k-SAT with Portfolio optimization

Example



Four feasible solutions!

Feasible solutions:

$[0,1,1,0,0]$, $[0,1,1,0,1]$, $[1,1,1,0,0]$, $[1,1,1,0,1]$

Create 5x5 matrix Σ, 5-dim return vector μ, budget b

Computed optimal weights (used QAOA)

$[0,3,7,0,0]$, $[0,2,3,0,5]$, $[3,2,5,0,0]$, $[1,2,3,0,4]$

Optimal expected returns:

$[26]$, $[17]$, $[27]$, $[21]$

Best solution:

$[1,1,1,0,0]$ **with expected return** $[27]$**!**

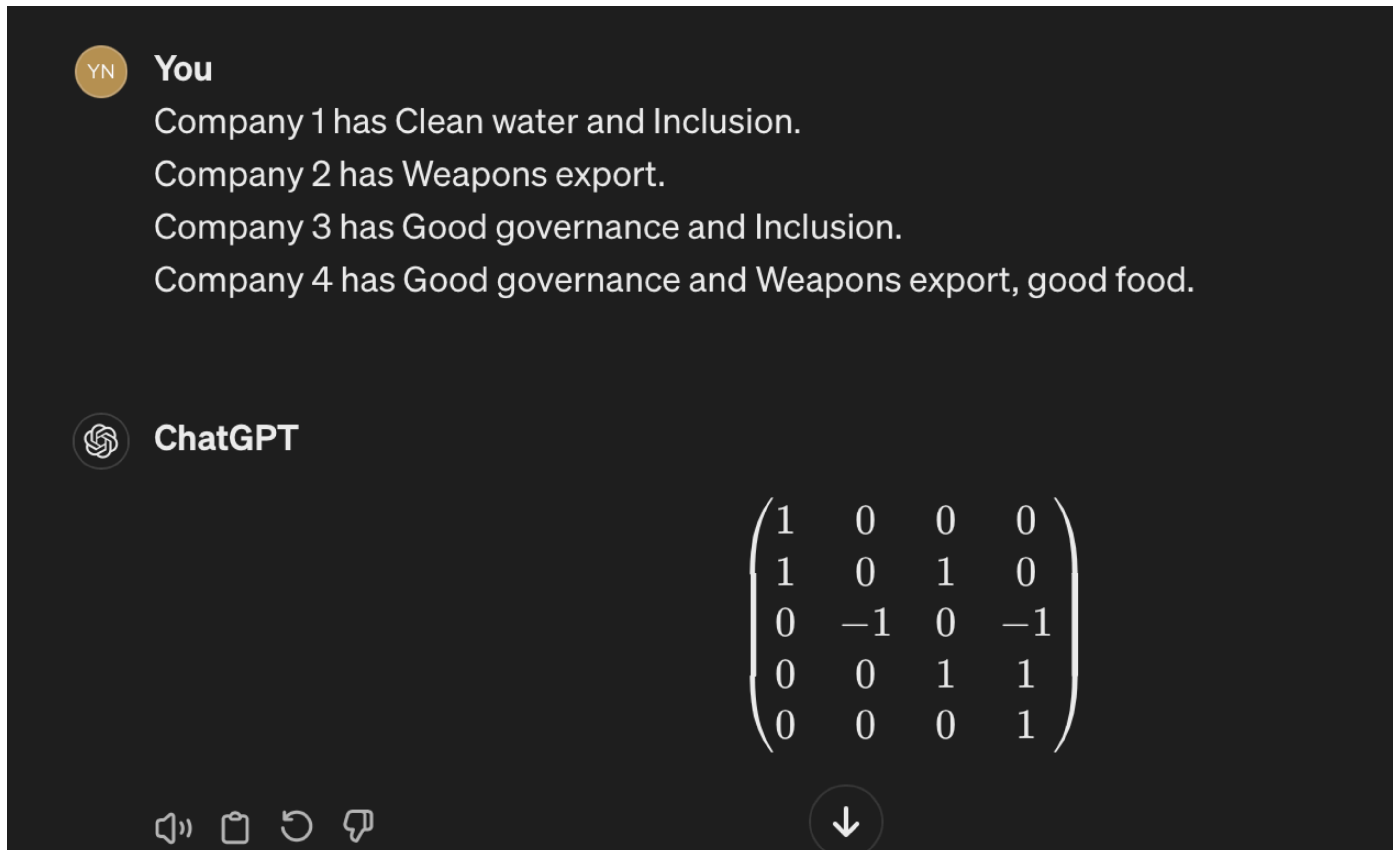
Outlook

SAT vs UNSAT

- **SAT:** Not allowed to violate any constraints
- **UNSAT:** Try to violate as few as possible constraints
- Often constraints are not violently enforced, have some free space
- Possibly include interplay between financial aspects (portfolio optimization) and trying to fulfil as many constraints as possible

Integrating LLM

- Bridge the gap between **natural language** and **k-SAT** problem definition



Thanks for listening!