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
5. **Exploring** the feasible solutions
6. **Outlook:** UNSAT, LLM's, and other applications

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 fulfill 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**

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 Mx$k$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

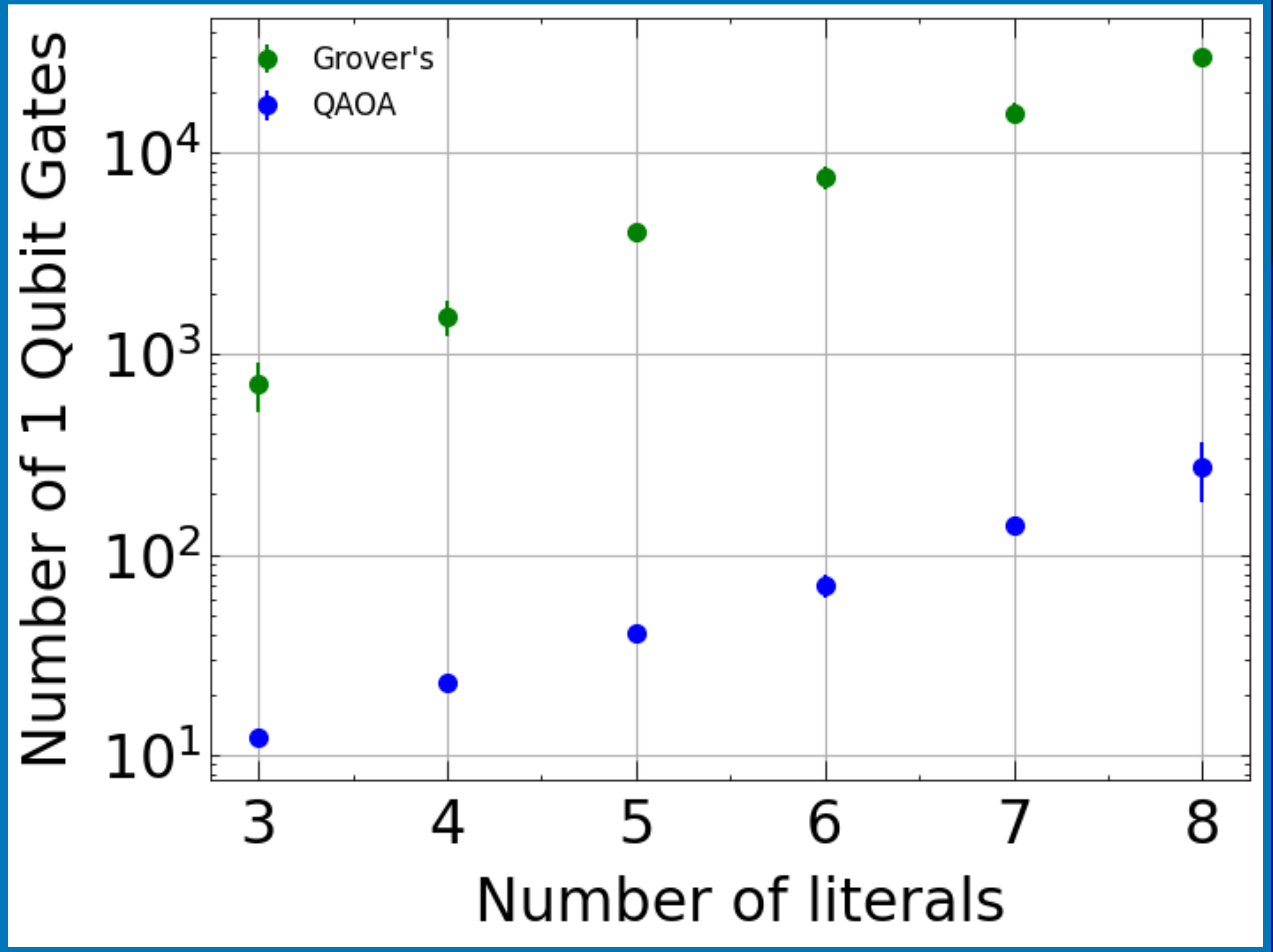
GROVER SOLVER

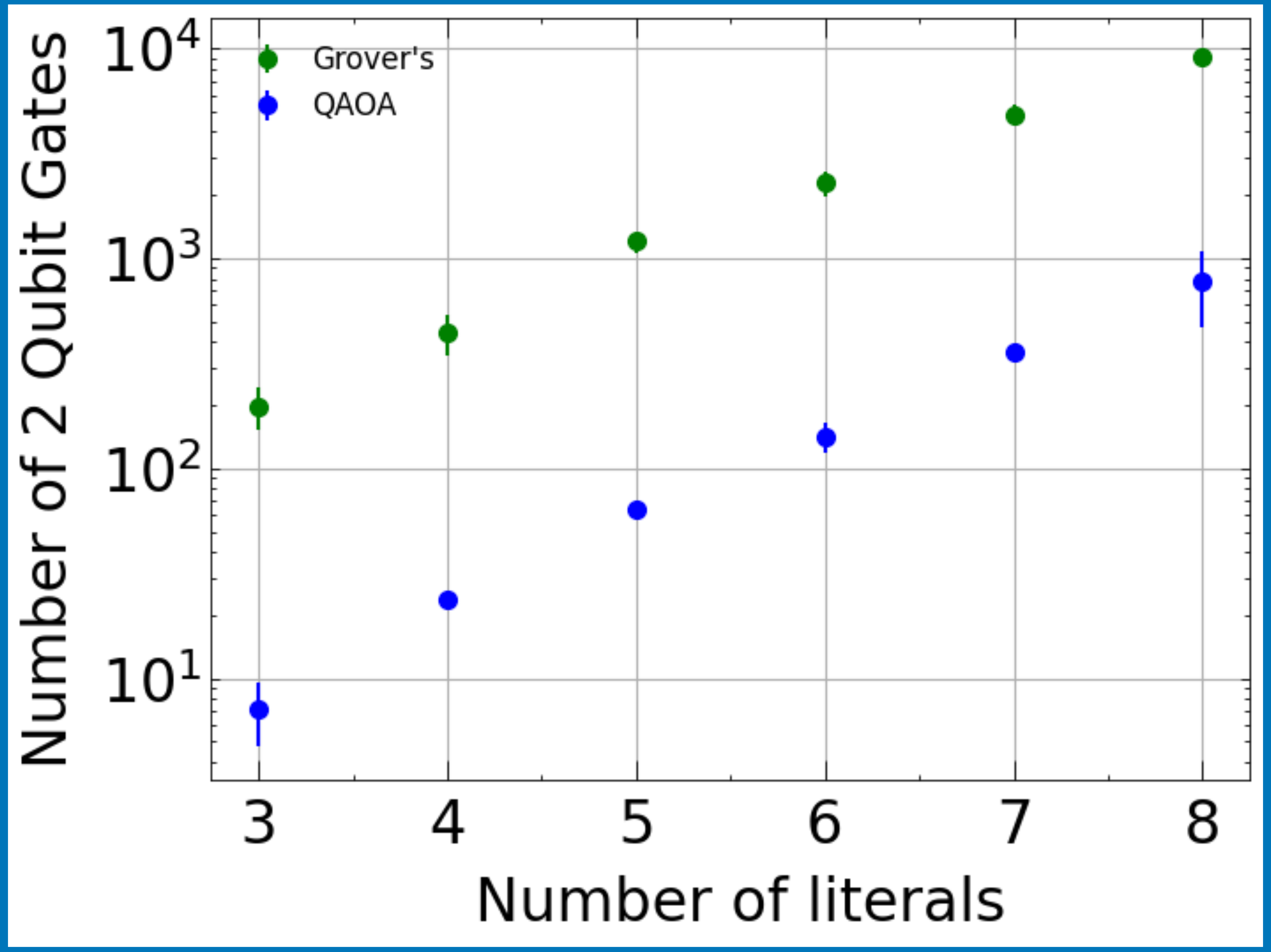
Remark: Included condition $x_1 \vee x_2 \ldots \vee x_k$ **to exclude trivial solutions** (no companies to invest in)

Number of gates needed per approach

k = # Literals (Companies)

M = # Clauses (Conditions)





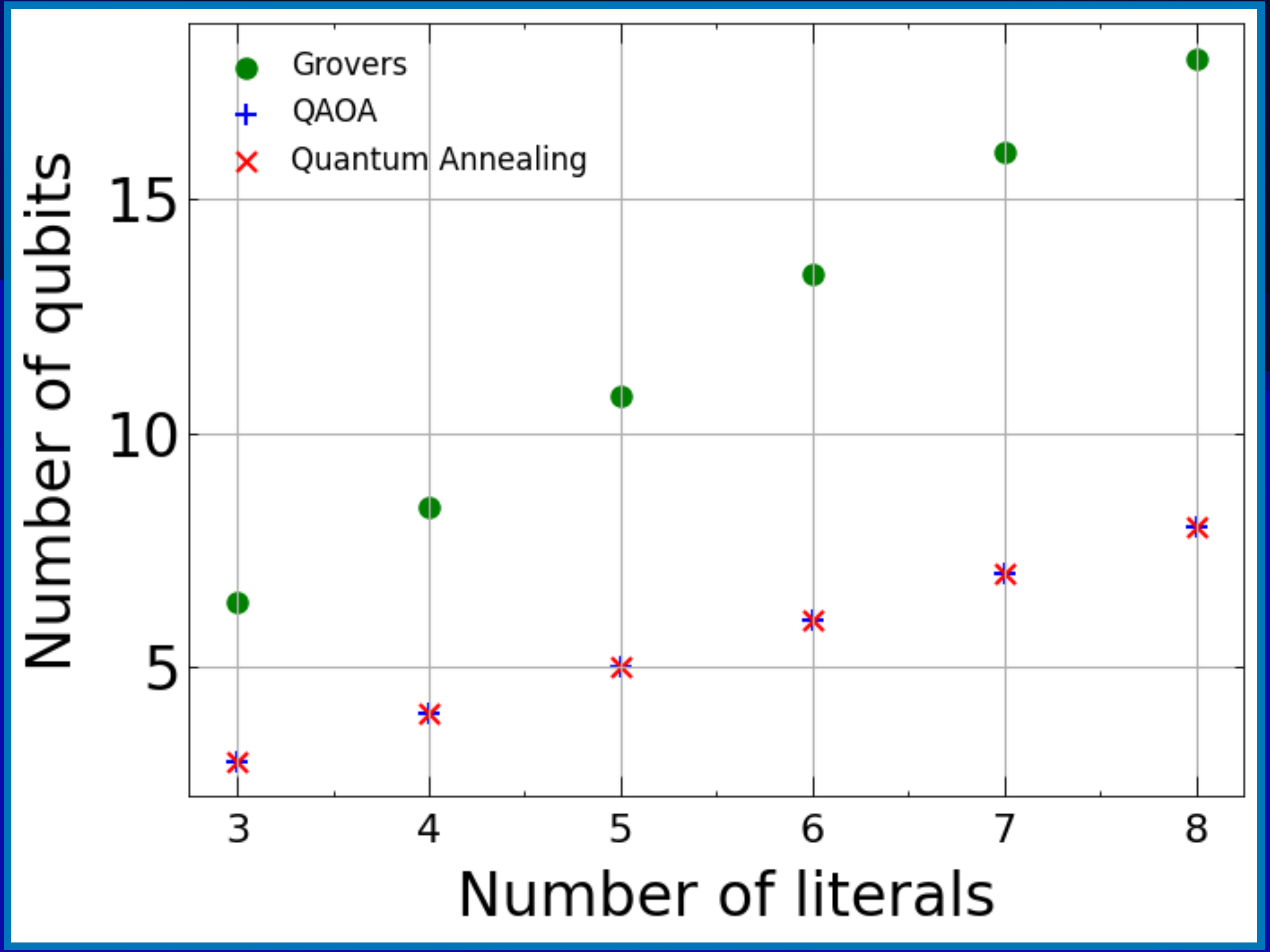
$k = M$

Quantum Annealing not included, doesn't use gates!

Number of qubits N needed per approach

k = # Literals (Companies)

M = # Clauses (Conditions)



$k = M$

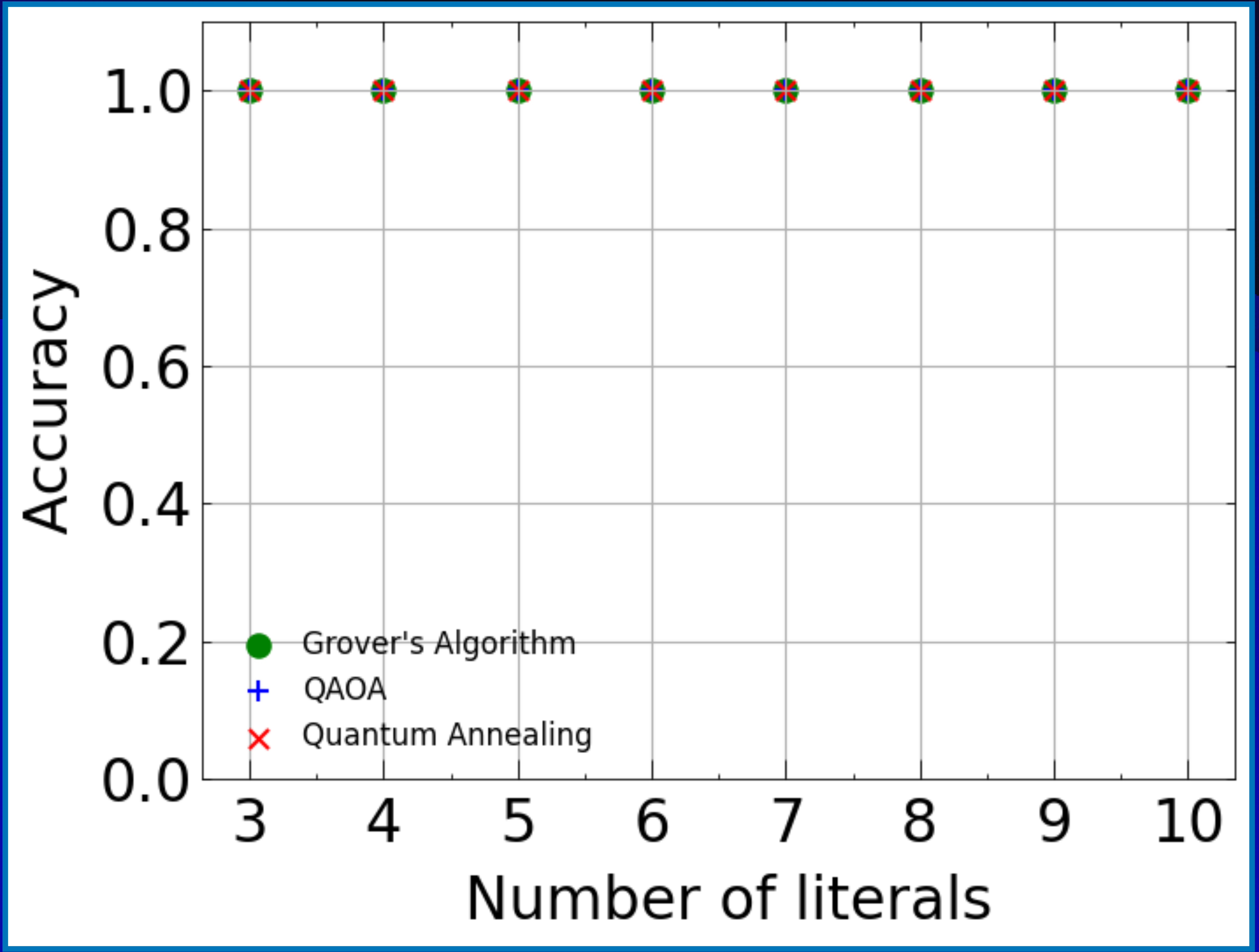
→ Quantum annealing and QAOA: $N = k$

→ Grover's algorithm: Linear scaling of N

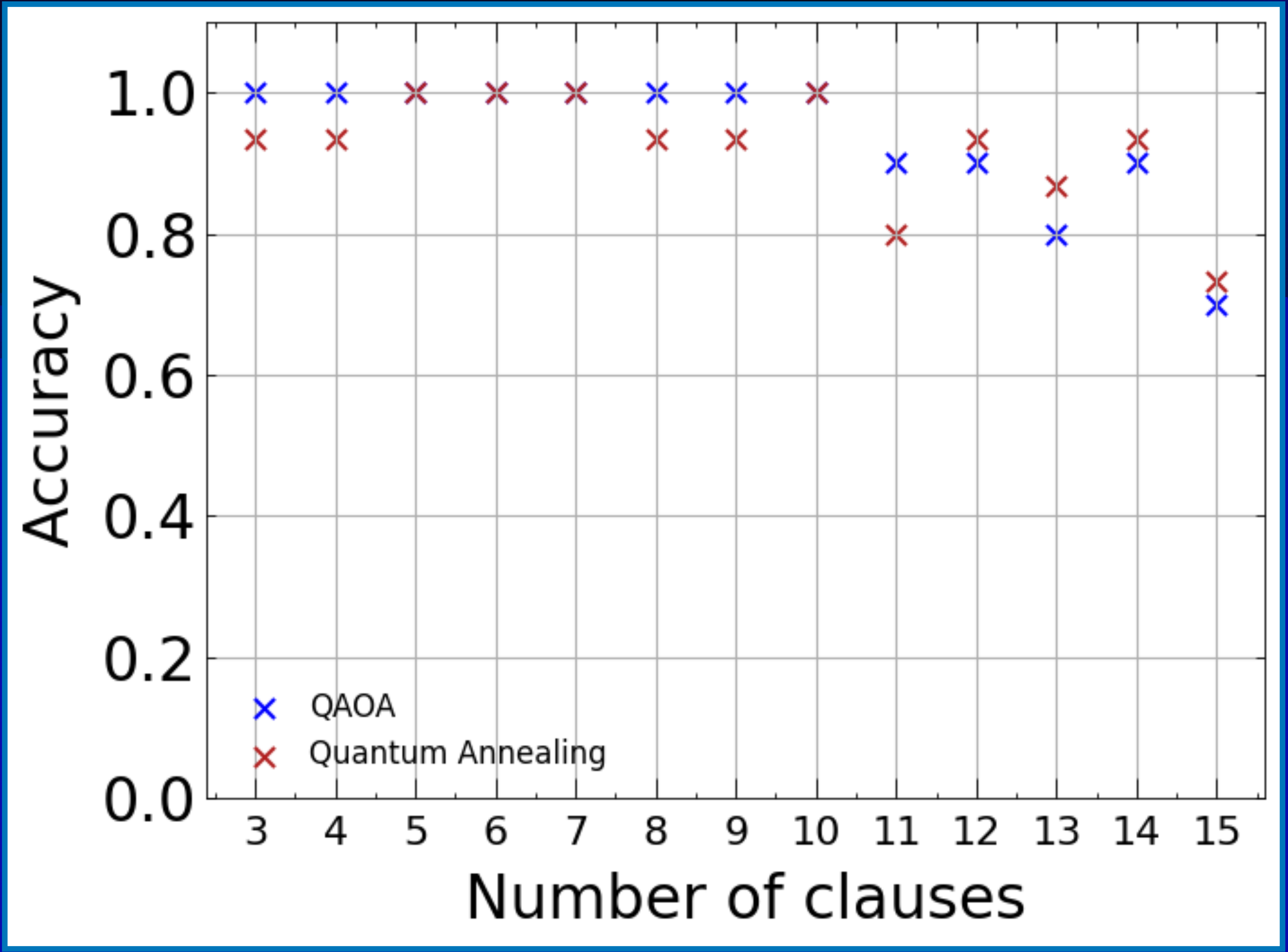
Accuracy analysis without noise

k = # Literals (Companies)

M = # Clauses (Conditions)

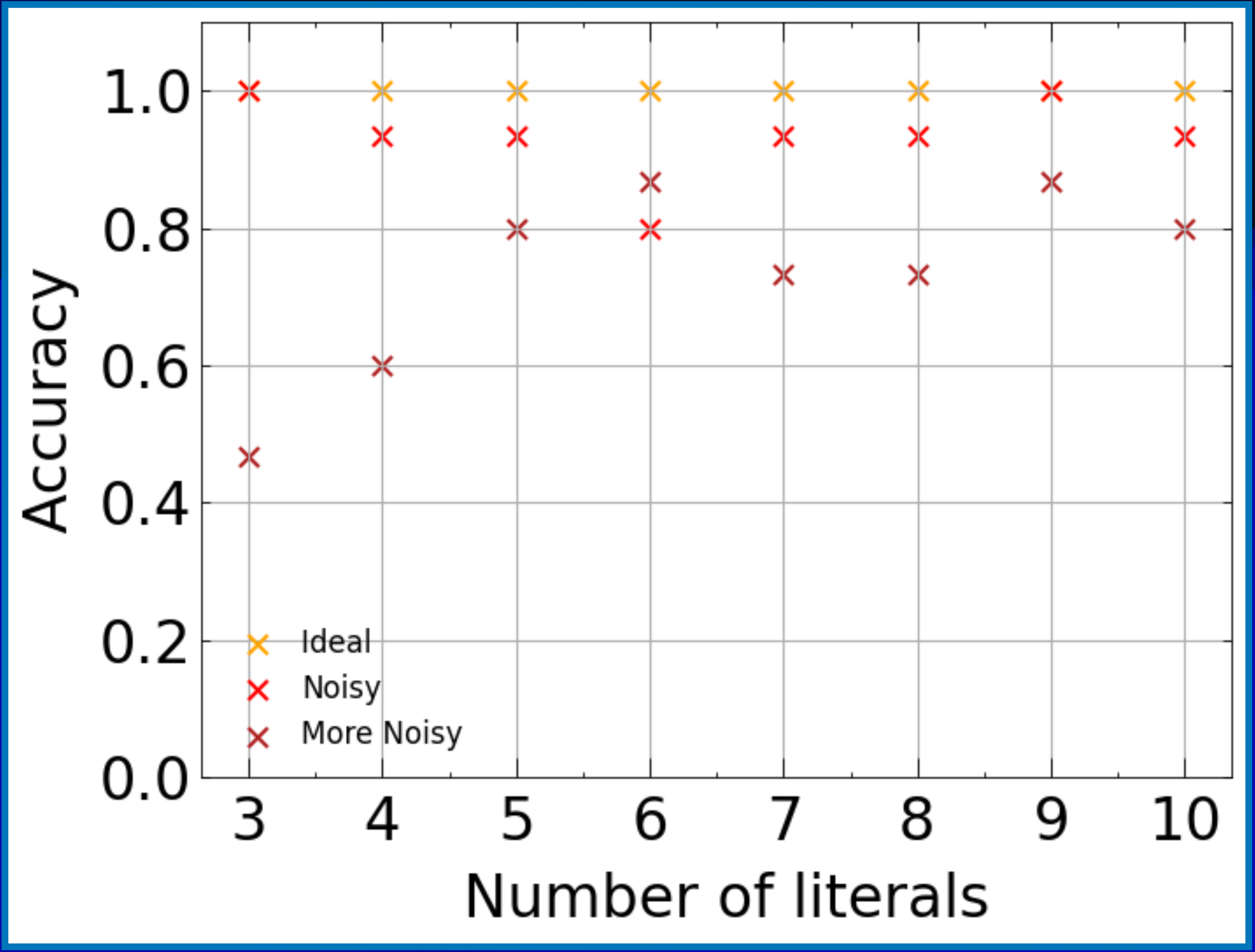


$k = M$



$k = 6,\ M$ variable

Accuracy analysis with noise for Quantum Annealing



$k = M$

M = # Clauses

k = # Literals

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**

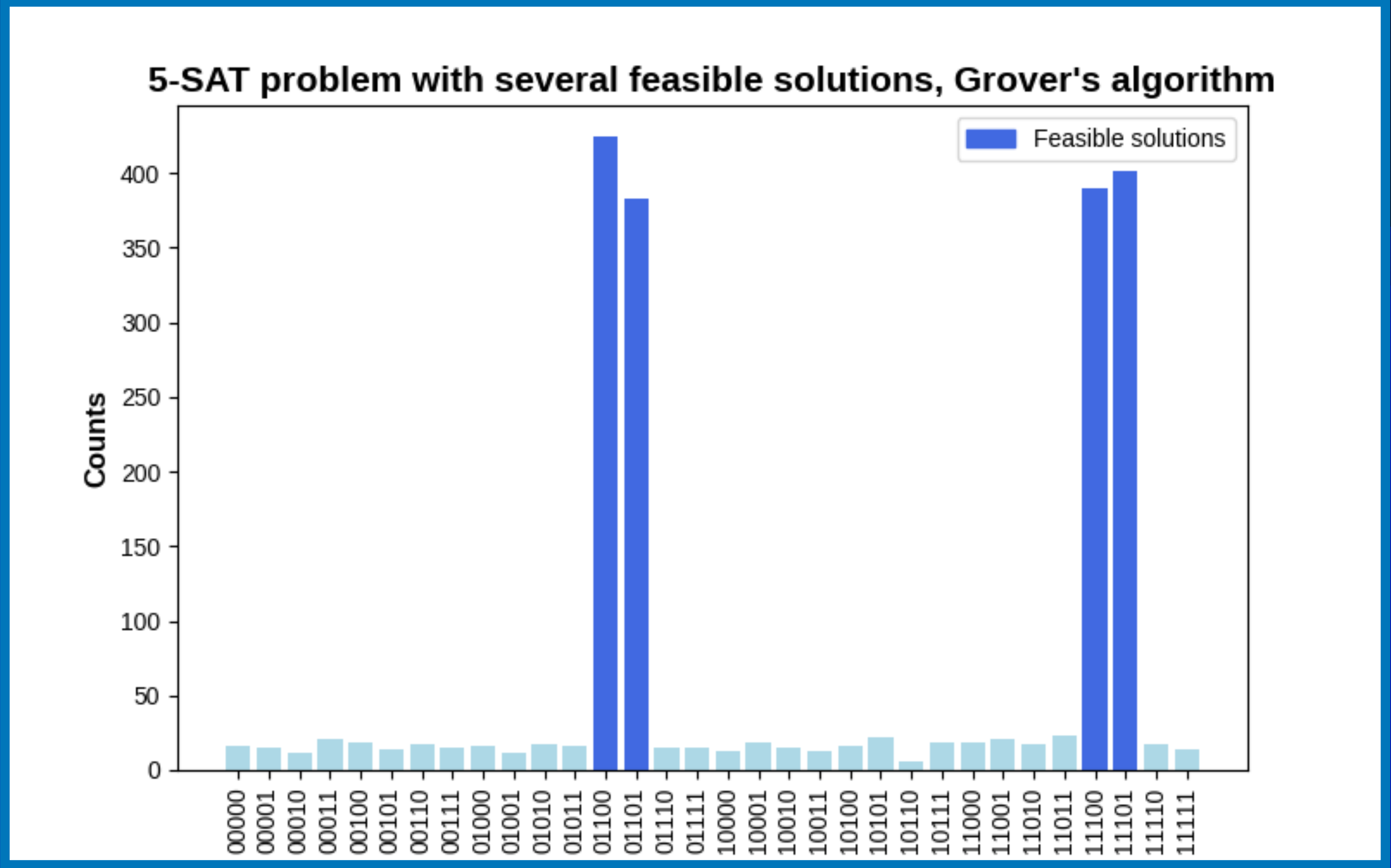
ALL SOLUTIONS
- All possible combinations of companies

FEASIBLE SOLUTIONS
- Companies omitting the ESG conditions

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

k-SAT with Portfolio optimization

An 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]!

Integrating LLM

You

Company 1 has Clean water and Inclusion.
Company 2 has Weapons export.
Company 3 has Good governance and Inclusion.
Company 4 has Good governance and Weapons export, good food.

ChatGPT

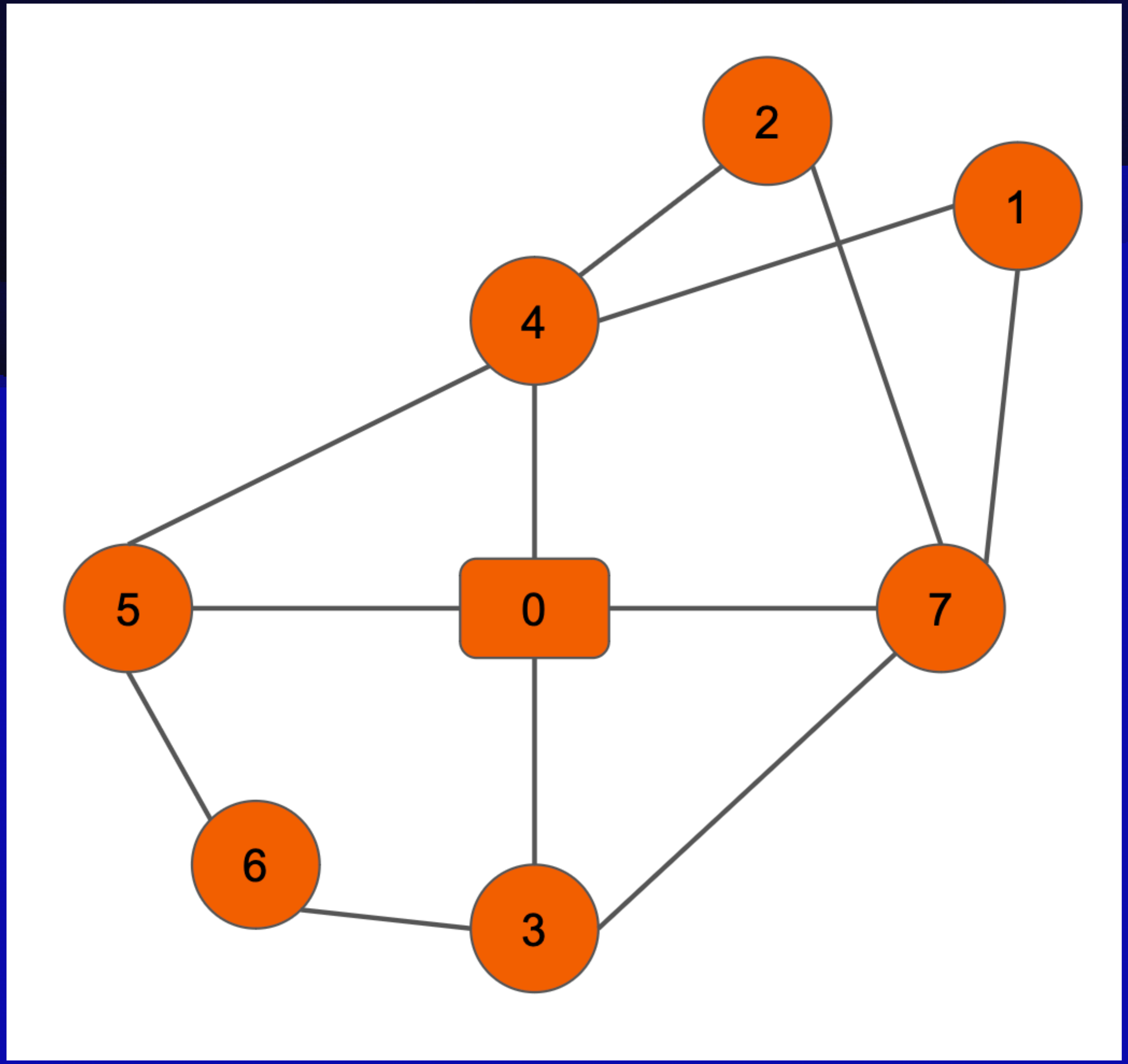
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Fully automated pipeline: From natural language to solution of optimization problem

Performed **prompt engineering** to bridge the gap between **natural language** and **k-SAT** problems

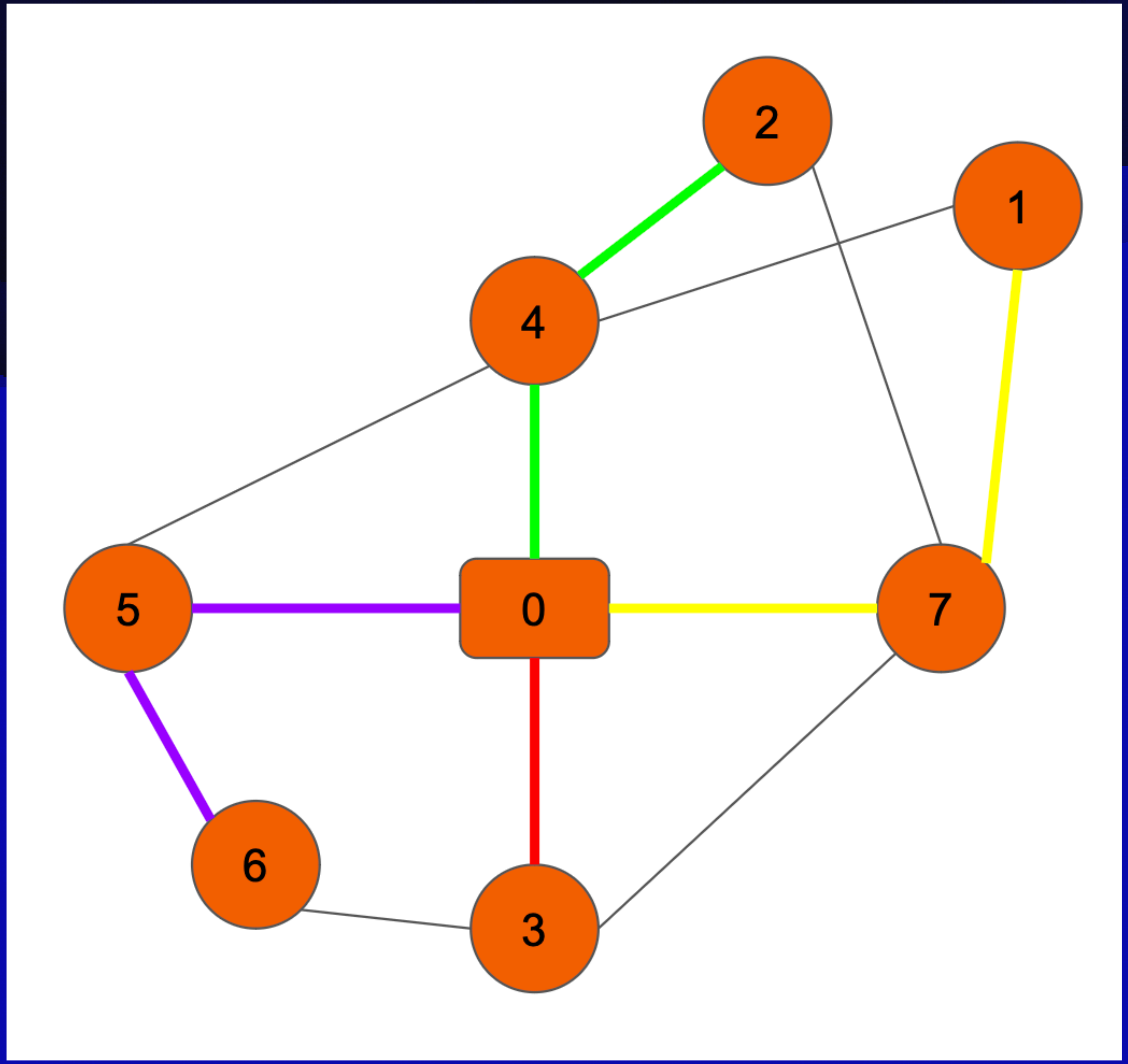
Other applications of k-SAT problems

- **Ressource allocation**
- Sudoku solving
- Planning meetings
- Job scheduling



Other applications of k-SAT problems

- **Ressource allocation**
- Sudoku solving
- Planning meetings
- Job scheduling



Conclusion

Grover's algorithm

- **Worst scaling in terms of resources (number of qubits, number of gates)**
- **Guaranteed to find solution**
- **Proven speedup!**
- **Not interesting for NISQ area, but way-to-go for fault-tolerant quantum computing!**

QAOA

- **Intermediate scaling in terms of resources (number of qubits, number of gates)**
- **No guarantee to find correct solution**
- **No proven speedup!**
- **NISQ algorithm, might be applicable in near future**

Quantum Annealing

- **Good scaling in terms of complexity**
- **No guarantee to find correct solution**
- **No proven speedup!**
- **Hardware already available, most realistic in near future!**

Thank you!

Grover's algorithm

- Worst scaling in terms of resources (number of qubits, number of gates)
- Guaranteed to find solution
- Proven speedup!
- Not interesting for NISQ area, but way-to-go for fault-tolerant quantum computing!

QAOA

- Intermediate scaling in terms of resources (number of qubits, number of gates)
- No guarantee to find correct solution
- No proven speedup!
- NISQ algorithm, might be applicable in near future

Quantum Annealing

- Good scaling in terms of complexity
- No guarantee to find correct solution
- No proven speedup!
- Hardware already available, most realistic in near future!