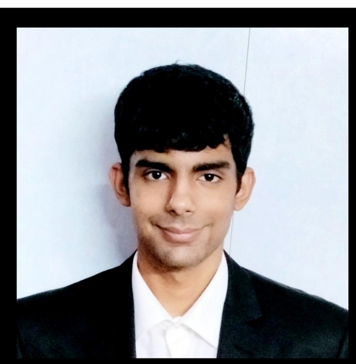


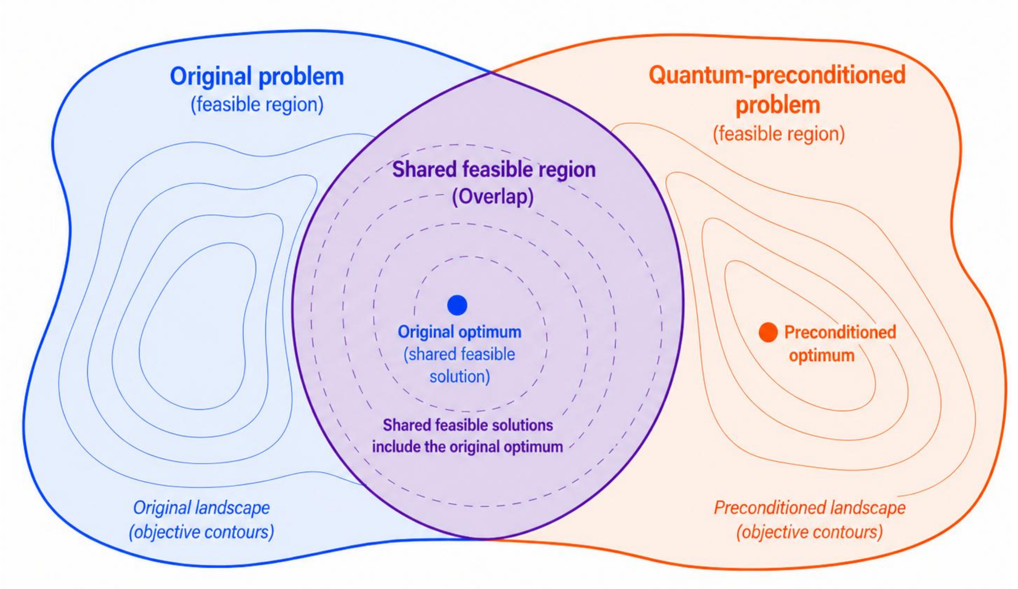
Investigating Quantum Preconditioning for Mixed-Integer Programming Solvers

Anurag Ramesh[a], Maxime Dupont[b], Bhuvanesh Sundar[b], David E. Bernal Neira[a]
[a]Davidson School of Chemical Engineering, Purdue University, West Lafayette, IN
[b]Rigetti Computing, Berkeley, CA

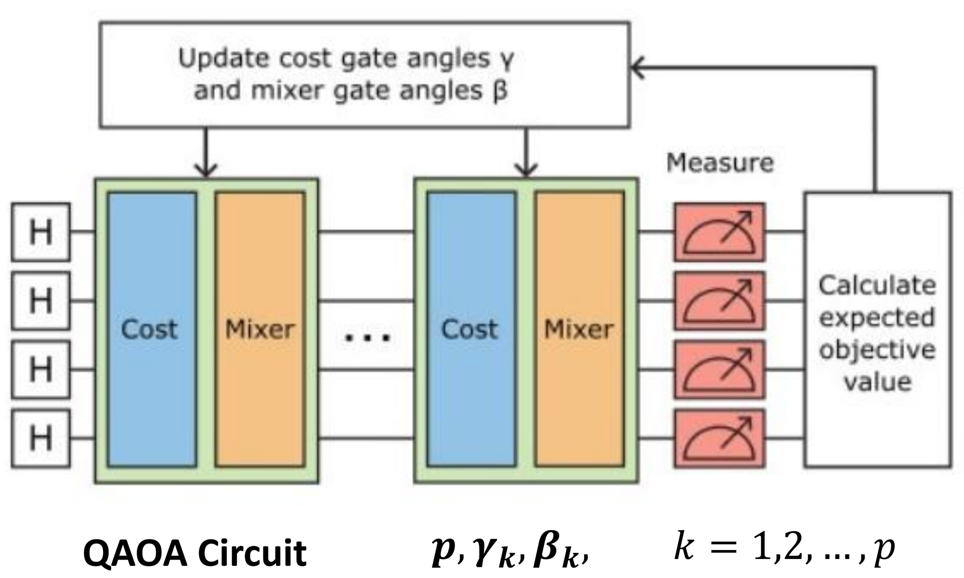
Motivation and Background

- Combinatorial optimization problems arise in a wide range of applications, including network design, circuit partitioning, and scientific computing.
- Modern **mixed-integer programming (MIP) solvers are highly effective** for such problems, but their **practical performance depends strongly** on **the problem structure** and **formulation** presented to the solver.
- Moreover, these solvers can **spend significant time certifying optimality** after good incumbents are found.
- To improve performance, we can perform a transformation called **"preconditioning[1]"** that seeks to transform the problem into a more suitable form for the solver.

Can quantum-generated information be used to reformulate or precondition an optimization problem so that classical solvers find high-quality solutions faster[3,6]?



Quantum Approximate Optimization Algorithm (QAOA)[4,5]



Problem Reformulation for QAOA

Integer Program:

$$\min_x c^T x + x^T W x \quad \text{s.t.} \sum_i x_i = s, \quad x_i \in \{0,1\}$$

⇕

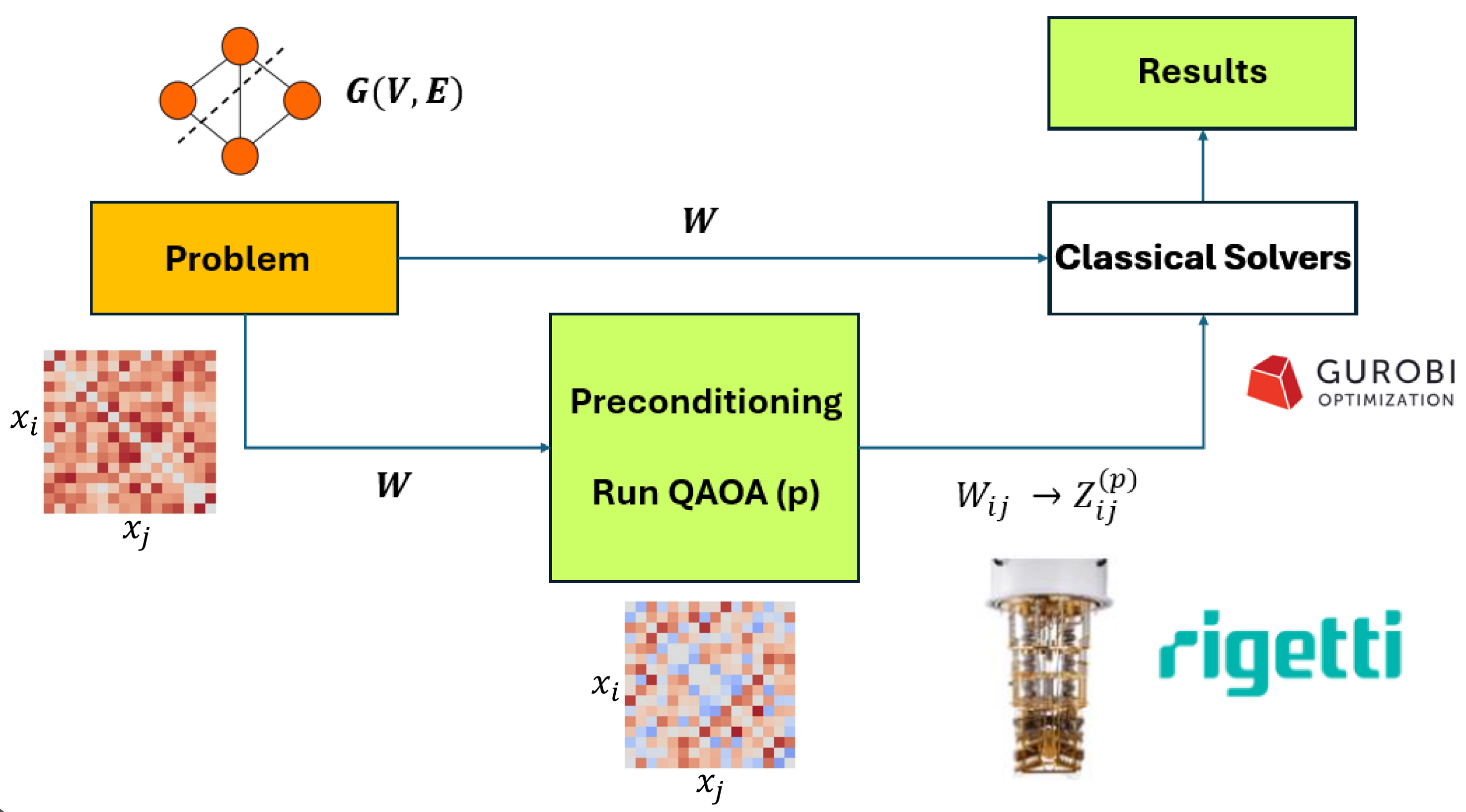
Quadratic Unconstrained Binary Optimization:

$$\min_x x^T \boldsymbol{Q} x + \rho\left(\sum_i x_i - s\right)^2 \quad \text{s.t.}\ x \in \{0,1\}^n$$

⇕

Ising Model:

$$\min_z \sum_{i,j} J_{ij} z_i z_j + \sum_i h_i z_i$$



Problem Formulation

- We study balanced graph bi-partitioning[7,8] on weighted complete graphs.
- Given an undirected graph $G = (V, E)$ with edge weights w_{ij},

$$\min_z \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(1 - z_i z_j)$$
$$\text{s.t.} \sum_{i\in V} z_i = 0, \quad z_i \in \{-1,+1\}, \quad \forall i \in V.$$
$$z_i = 2x_i - 1, \quad z \in \{-1,1\}, \quad x \in \{0,1\}$$

- Integer Program → Quadratic Unconstrained Binary optimization for Preconditioning:

$$\min_x w_{ij}(x_i + x_j - 2x_i x_j) + \rho\left(\sum_{i\in V} x_i - \frac{|V|}{2}\right)^2, \qquad s.t.\ x \in \{0,1\}^n$$

- Generate Correlations between decision variables using QAOA:

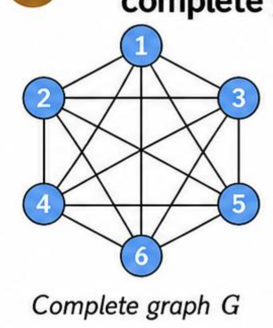
$$Z_{ij}^p = (\delta_{ij} - 1)\langle z_i z_j \rangle, \qquad \textit{where } \langle . \rangle \textit{ represents expectation value}$$

- Quantum Preconditioned Formulation:

$$\min_z \frac{1}{2} \sum_{(i,j)\in E} Z_{ij}(1 - z_i z_j)$$
$$\text{s.t.} \sum_{i\in V} z_i = 0, \quad z_i \in \{-1,+1\}, \quad \forall i \in V.$$

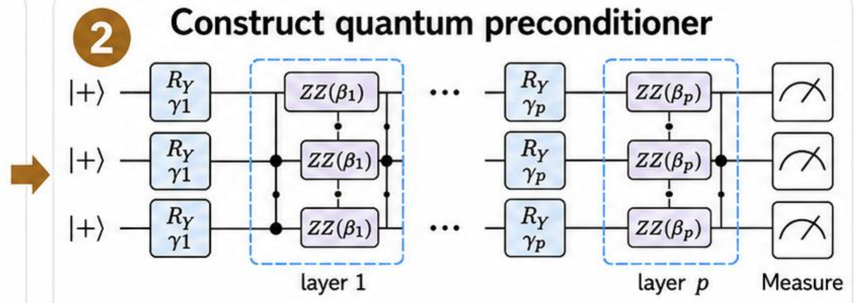
Methods

1 Generate weighted complete graph instances



- $G = (V, E), N = |V|$
- Edge weights $w_{ij} \sim U(0, 2)$
- Multiple random seeds for each N

2 Construct quantum preconditioner

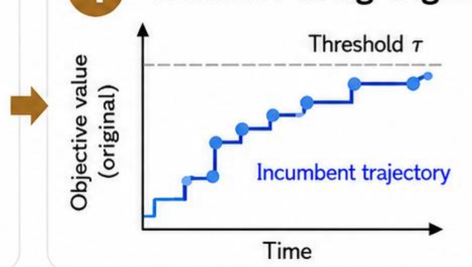


- Use state-vector quantum circuit simulations for this experiment
- Run QAOA with depth p
- Compute two-point correlations $Z_{ij}^{(p)}$
- The same correlations can also be generated using a quantum computer

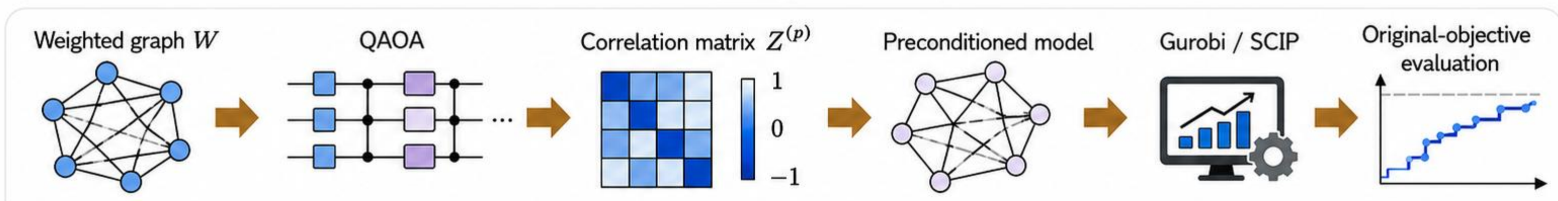
3 Solve preconditioned optimization problem

- Baseline: original graph weights w_{ij}
- Preconditioned: quantum-derived weights $Z_{ij}^{(p)}$
- Solvers: Gurobi / SCIP
- Penalty values swept over a grid

4 Evaluate using original objective

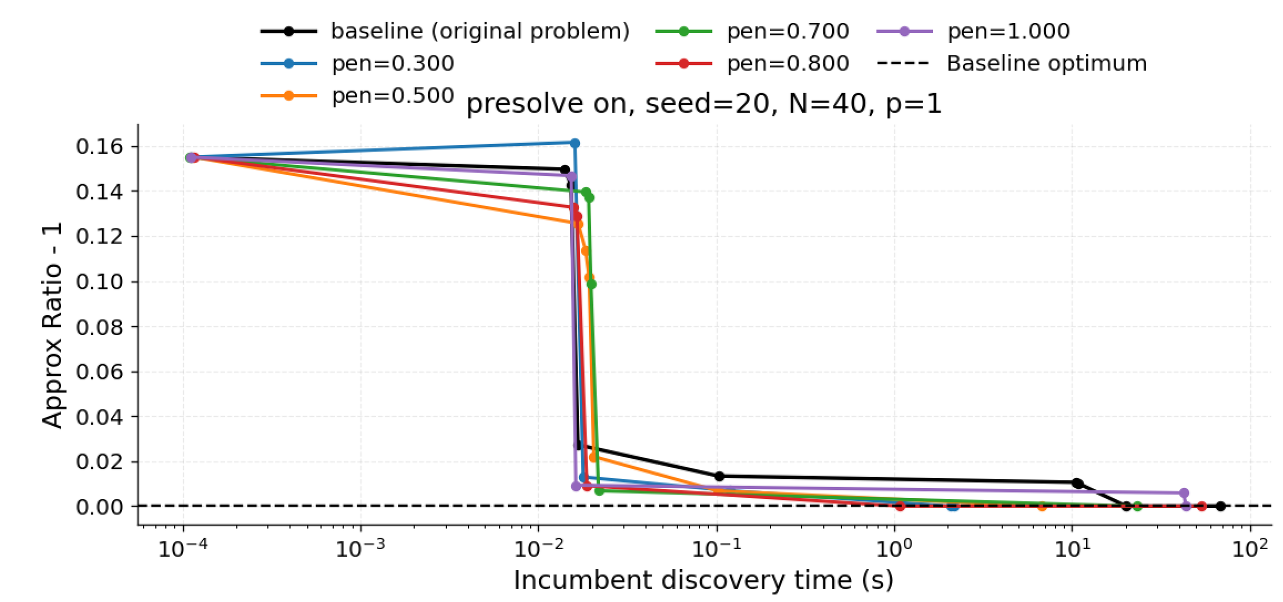


- Record incumbent solutions using solver callbacks
- Re-score every incumbent using the original objective

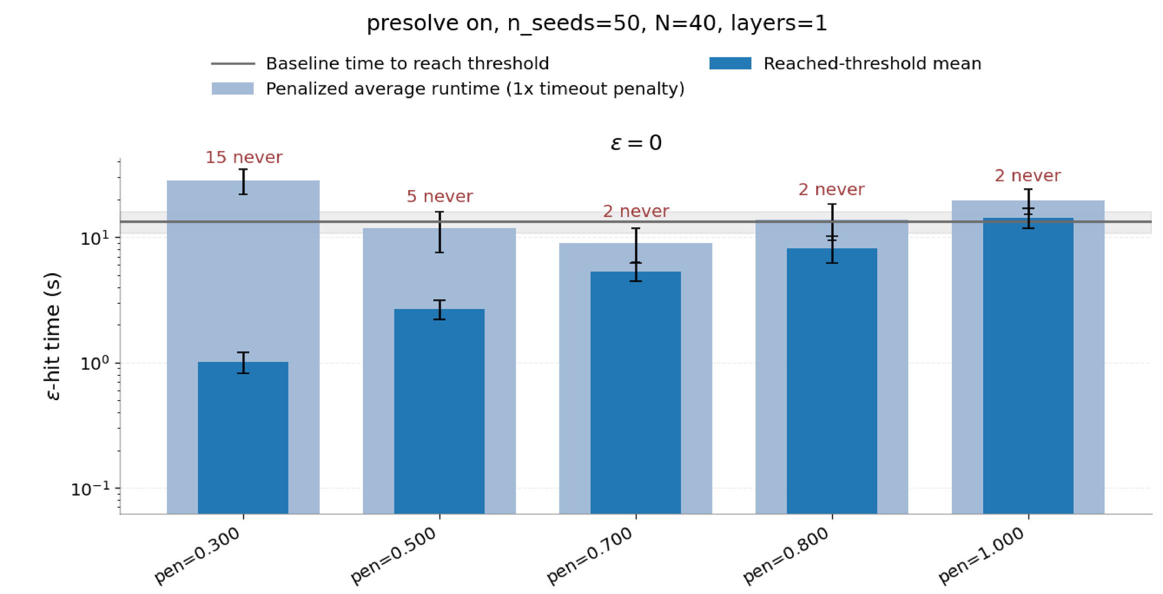


Results

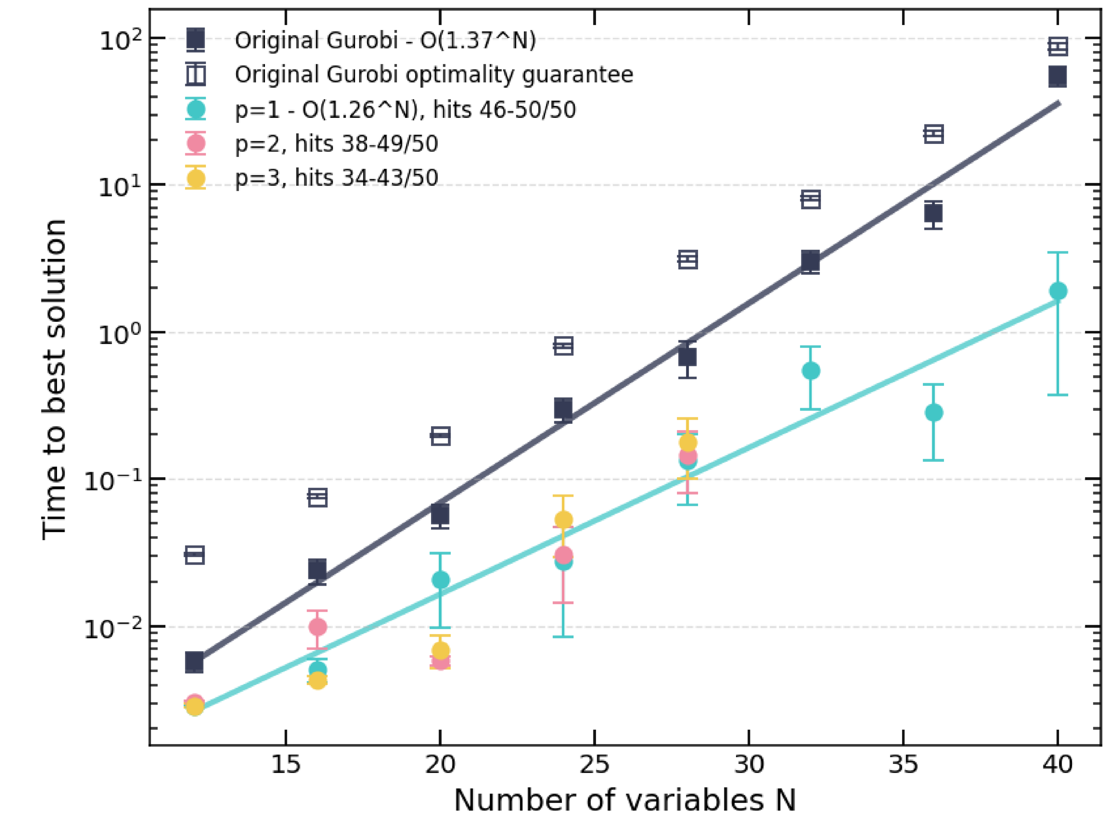
1. Preconditioning changes incumbent trajectories



2. ϵ −hit runtime (N = 40)



1. Quantum Preconditioning improves empirical scaling behavior of the MIP solver



Performance Metrics

Metric	Equation
Approx ratio	$AR = \frac{f_{orig}}{f^*}$
Time-to-best	$t_{best} = \min\{\, t : f_{orig}(t) = f^* \}$
ε-hit time	$t_\varepsilon = \min\{\, t : f_{orig}(t) \le f^* + \lvert f^* \rvert \varepsilon \}$
Penalized average time	$T_{pen} = T_{run} + \alpha T_{baseline}$

Notation:

$f_{orig}(z)$: original-objective value at solution z;

f^*: optimal/best-known value of the original problem;

$T_{baseline}$: runtime of the baseline/original solve.

Conclusions & Future Work

- **Quantum preconditioning** may **accelerate incumbent solution discovery**
- The **benefit** depends **strongly** on selected **penalty** strengths $\boldsymbol{\rho}$ and **QAOA depth** $\boldsymbol{p}$..
- **Preconditioning** can **improve the solver's search trajectory** by helping it discover high-quality incumbents earlier, but this **benefit** must **be weighed against** the **overhead** of **generating** the **correlation matrices**.
- Simulated correlations show promise, but hardware noise, sampling cost, and scalability remain open challenges.
- **Future work:** Extend to other constrained problems,, test hardware-generated correlations, and compare with classical preconditioners.

References

1. A. J. Wathen, Preconditioning, Acta Numer., 24 (2015), pp. 329–376, https://doi.org/10.1017/231/S0962492915000021
2. M. Dupont, T. Oberoi, and B. Sundar, Optimization via quantum preconditioning, Phys.225 Rev. Appl., 24 (2025), p. 044013, https://doi.org/10.1103/9prw-684p, https://link.aps.org/226/doi/10.1103/9prw-684p.
3. https://www.gurobi.com/documentation/current/refman/index.html
4. Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. arXiv:1411.4028, 2014.
5. Tameem Albash and Daniel A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys., 2018
6. Rieffel, E.G. and Polak, W.H., 2011. Quantum computing: A gentle introduction. MIT Press.
7. Wolsey, L. A., Integer programming. John Wiley & Sons,. 2020.
8. https://patterns.eecs.berkeley.edu/?page_id=571

Acknowledgement.
We gratefully acknowledge the Center for Quantum Technologies at Purdue University for their financial support of this work.. We also thank Gurobi Optimization (version 13.0) for providing a free academic license,.