MIT 6.2400: Introduction to Quantum Systems Engineering

Quantum Approximate Optimization for LUT Cut Selection in Logic DAGs

Dylan Bradford

Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02142 [a)]

(Dated: December 10, 2025)

Mapping Boolean logic networks to FPGA lookup tables (LUTs) requires selecting a consistent set of k-feasible cuts across a directed acyclic graph (DAG). This selection process is inherently combinatorial: each node may have several valid cuts, and these choices interact globally through overlap, depth, and fanout constraints. In this work, we formulate cut selection as a Quadratic Unconstrained Binary Optimization (QUBO) problem and study how the Quantum Approximate Optimization Algorithm (QAOA) performs on these highly structured instances.

We develop three progressively stronger QUBO models: (A) a baseline that enforces the "exactly one cut per node" legality constraint, (B) a depth-biased model that penalizes cuts rooted at deeper internal nodes, and (C) a duplication-aware model that further penalizes combinations of cuts that replicate internal logic. Using the standard substitution $x_i \mapsto (1 - Z_i)/2$, we map each QUBO into an Ising Hamiltonian suitable for QAOA. Because Qiskit's QAOA implementation proved too slow for even moderate instances ($n = 11$), we built a custom statevector simulator implementing the cost and mixer unitaries, depth-p ansatz, and classical parameter optimization.

We observe smooth, structured energy landscapes and strong probability concentration on low-energy solutions for small QAOA depth. Increasing p improves ground-state overlap, particularly for the more degenerate Model C. Finally, we extend our method to a reduced 2-bit ripple-carry adder, showing that QAOA preserves useful probability–energy structure even when only a subset of variables is simulated. These results provide insight into the potential and current limitations of quantum variational algorithms for hardware-motivated combinatorial optimization problems.

1. Introduction

Field-programmable gate arrays (FPGAs) implement Boolean functions using fixed-size lookup tables (LUTs). A central step in LUT-based FPGA technology mapping is cut selection: for each internal node of a logic network represented as a directed acyclic graph (DAG), the mapper must select a k-feasible cut whose leaves can be implemented as a single k-input LUT.[1] Modern mappers such as ABC apply sophisticated dynamic-programming and structural-choice heuristics, and ongoing work continues to refine these choices for better area and timing trade-offs.[2] Nevertheless, the underlying problem remains a highly structured combinatorial optimization task.

On the quantum side, the Quantum Approximate Optimization Algorithm (QAOA)[3] is a variational algorithm that approximates the ground state of an Ising Hamiltonian by alternately applying a problem-specific cost unitary and a generic mixer unitary, with parameters tuned by a classical optimizer. QAOA has been extensively reviewed and benchmarked on canonical graph problems such as MaxCut,[4] but there are fewer case studies on engineering-motivated, structured instances that resemble real CAD workloads.

In this project, we connect these two worlds. We formulate LUT cut selection as a Quadratic Unconstrained Binary Optimization (QUBO) problem, following the general framework of Glover et al.[5], and then map this QUBO to a Pauli-Z Hamiltonian suitable for QAOA. We focus on small, fully analyzable examples—a 1-bit full adder and a reduced 2-bit ripple-carry adder—and use them as a sandbox to study how QAOA behaves on structured logic-DAG instances: how probability mass is distributed over low-energy configurations, how this changes with cir-

[a)] **Electronic addresses:** {dylanjb,dylanjb}@mit.edu

cuit depth p, and how degeneracies and penalty terms affect the landscape.

Contributions

This work makes the following contributions:

1. We formulate FPGA cut selection as a family of QUBO models (Models A–C) that encode exactly-one-cut legality, depth preference, and duplication penalties in a DAG-based LUT mapping setting, building on classical mapping ideas in Refs. 1 and 2.

2. We convert these QUBOs into diagonal Ising Hamiltonians using the standard binary-to-spin substitution $x_i = (1 - Z_i)/2$ from the QUBO literature,[5] and verify correctness by brute-force enumeration of all 2^n bitstrings.

3. Motivated by performance issues with black-box libraries, we implement a custom QAOA statevector simulator (cost and mixer unitaries, depth-p ansatz, classical optimization, and measurement) tailored to these Hamiltonians.

4. We characterize QAOA behavior on structured problems through energy landscapes, probability–energy relationships, and ground-state overlap for both a 1-bit full adder and a reduced 2-bit ripple-carry adder, relating the trends we observe to expectations from QAOA theory and benchmarks in Ref. 4.

5. We discuss limitations and scaling challenges, and outline how such structured instances could serve as realistic benchmarks for future quantum–classical optimization workflows.

2. Background

2.1. Logic Networks, DAGs, and LUT Cuts

A Boolean circuit can be represented as a directed acyclic graph (DAG), where each internal node corresponds to a logic operation and edges represent signal flow. In LUT-based FPGA mapping, a k-feasible cut of a node is a set of leaf inputs whose transitive fan-in contains at most k distinct input signals. Each such cut can be realized as one k-input LUT implementing the node's function in terms of those leaves.[1]

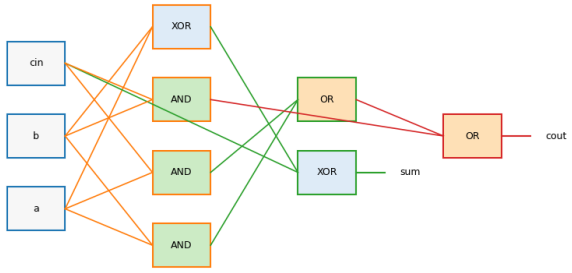

FIG. 1: Logic DAG for the 1-bit full adder used in this work, with internal nodes labeled by gate type and example k-feasible cuts highlighted. Each cut corresponds to a potential LUT implementation rooted at that node.

A mapper must choose exactly one cut for every internal node, subject to the structure of the DAG. Different choices change how much logic is replicated, how many LUT levels a path traverses, and how many times an intermediate signal is re-used, producing trade-offs in area, depth, and fanout. Recent work on mixed structural-choice optimization further highlights how many competing mapping options exist even for a fixed netlist.[2]

2.2. QUBO Models

Quadratic Unconstrained Binary Optimization (QUBO) provides a standard way to encode combinatorial problems in terms of binary variables and quadratic penalties.[5] Let $x_{i,c} \in \{0, 1\}$ indicate whether cut c is chosen for node i, and let C_i be the set of cuts of node i. We define three models with increasing realism:

- **Model A (exactly-one constraint)** enforces that each node selects exactly one cut. For every node i we penalize deviations from one chosen cut:

$$E_A(x) = \sum_i \left(\sum_{c \in C_i} x_{i,c} - 1 \right)^2 . \tag{1}$$

  This defines a legal baseline: any ground state of E_A corresponds to a cut assignment in which every node has exactly one cut.

- **Model B (depth penalty)** biases the mapper toward shallower cuts. We assign each cut a scalar $d_{i,c}$ proportional to its logic level or

distance from the primary inputs, and add a linear term

$$E_{\text{depth}}(x) = \sum_{i,c} d_{i,c}\, x_{i,c}. \tag{2}$$

The full Model B objective is then

$$E_B(x) = E_A(x) + \lambda_{\text{depth}}\, E_{\text{depth}}(x), \tag{3}$$

where λ_{depth} controls how aggressively we penalize deep, internal cuts. This model is intended to approximate depth-oriented mapping heuristics used to reduce critical path delay.

- **Model C (duplication penalty)** discourages cut combinations that duplicate internal logic and increase LUT count. Let $\mathcal{P}$ index pairs of cuts (i, c) and (j, d) whose simultaneous selection would replicate some subgraph. We add a quadratic penalty

$$E_{\text{dup}}(x) = \sum_{((i,c),(j,d))\in\mathcal{P}} p_{(i,c),(j,d)}\, x_{i,c} x_{j,d}, \tag{4}$$

where $p_{(i,c),(j,d)} > 0$ encodes the strength of the duplication cost. The full Model C objective is

$$E_C(x) = E_A(x) + \lambda_{\text{dup}}\, E_{\text{dup}}(x). \tag{5}$$

In practice this produces a small manifold of degenerate ground states corresponding to different low-duplication mappings.

2.3. QUBO to Ising Hamiltonian

A generic QUBO can be written as

$$E(x) = x^{\mathsf{T}} Q\, x + \text{const},$$

where $x \in \{0,1\}^n$ and Q is a symmetric matrix. Following the standard binary-to-spin mapping described in Ref. 5, we define

$$x_i = \frac{1 - Z_i}{2},$$

where Z_i is the Pauli-Z operator acting on qubit i.

Substituting into the QUBO and collecting coefficients yields a diagonal Ising Hamiltonian

$$H_{\text{cost}} = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j + \text{const}.$$

The ground state(s) of H_{cost} correspond exactly to the optimal cut-selection assignments.

2.4. Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA)[3,4] prepares a variational quantum state by alternating between a cost unitary and a mixer unitary. Starting from the uniform superposition $|+\rangle^{\otimes n}$, the depth-p ansatz is

$$\left|\psi(\vec{\gamma},\vec{\beta})\right\rangle = \prod_{l=1}^{p} U_B(\beta_l)\, U_C(\gamma_l)\; |+\rangle^{\otimes n},$$

where

$$U_C(\gamma) = e^{-i\gamma H_{\text{cost}}}, \qquad U_B(\beta) = e^{-i\beta \sum_i X_i},$$

and X_i is the Pauli-X operator on qubit i.

A classical optimizer updates the parameters $(\vec{\gamma},\vec{\beta})$ to minimize the expected cost $\left\langle\psi(\vec{\gamma},\vec{\beta})\right| H_{\text{cost}} \left|\psi(\vec{\gamma},\vec{\beta})\right\rangle$. As emphasized in recent reviews,[4] the quality of QAOA's approximate solution depends both on problem structure and on how well this classical search navigates the energy landscape defined by (γ, β).

3. Methods

3.1. Constructing the QUBO

For each node in the full-adder DAG, we enumerate all k-feasible cuts ($k = 3$). Each cut corresponds to a binary variable, yielding 11 variables in Models B and C. Penalties are computed based on internal-node exposure (Model B) and shared-subgraph duplication (Model C).

3.2. Hamiltonian Conversion

We substitute $x = (1 - Z)/2$ into the QUBO and collect coefficients into diagonal Pauli terms Z_i and $Z_i Z_j$. We verify correctness by computing the QUBO energy and Hamiltonian eigenvalues for all 2^{11} bitstrings and confirming exact agreement.

3.3. Manual QAOA Simulator

Qiskit's QAOA implementation failed to converge within reasonable time, so we constructed our own simulator:

- **State representation**: full 2^n statevector.
- **Cost layer**: $U_C(\gamma) = \exp(-i\gamma H_{\text{cost}})$, implemented by multiplying phases of computational basis states.
- **Mixer layer**: $U_B(\beta) = \exp(-i\beta \sum_i X_i)$, implemented via tensor products.
- **Depth**: experiments with $p = 1, 2$.
- **Optimization**: Nelder–Mead search over (γ, β).

3.4. Classical Optimization

At each optimization step, we compute the expected energy $\langle\psi|H_{\text{cost}}|\psi\rangle$ and update parameters. Once converged, we sample probability distributions over computational basis states.

4. Results

4.1. Energy Landscape for Model B

Figure 2 shows the expected energy $\langle\psi(\gamma,\beta)|H_{\text{cost}}|\psi(\gamma,\beta)\rangle$ for Model B at depth $p = 1$ over a grid of parameter values. The landscape exhibits a smooth valley leading to a well-defined minimum, around which the Nelder–Mead optimizer converges. This is consistent with the structure of Model B: the dominant cost term is a nearly linear depth penalty on individual cuts, so nearby parameter settings tend to change the expected energy in a gradual, almost convex fashion rather than creating a highly rugged landscape.

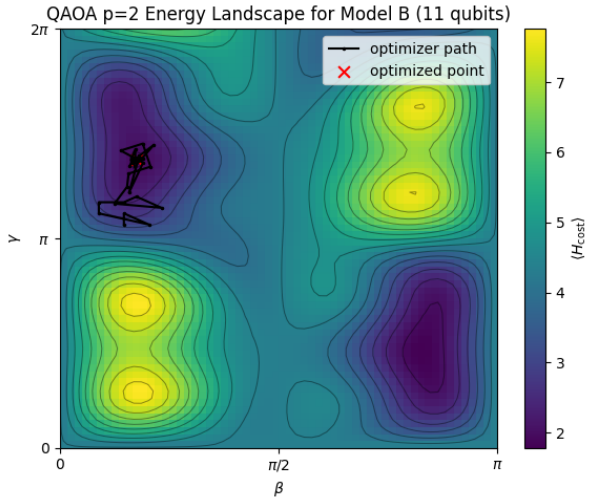

FIG. 2: Expected energy landscape for Model B at depth $p = 1$ as a function of (γ, β). Darker regions correspond to lower energy. The overlaid path shows the trajectory of the classical optimizer.

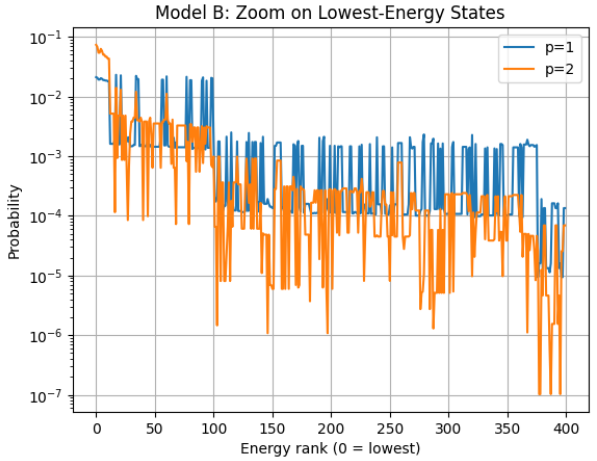

FIG. 3: Sampling probability versus classical energy rank for Model B. Depth $p = 1$ and $p = 2$ show increasing concentration of probability on low-energy states as circuit depth increases.

4.2. Probability vs. Energy Rank for Model B

Figure 3 plots the sampling probability $P(z)$ versus classical energy rank (from lowest to highest) for depth $p = 1$ and $p = 2$. For $p = 1$, low-energy states already receive noticeably higher probability than high-energy states, although the distribution is relatively broad. Increasing the depth to $p = 2$ sharpens the downtrend: probability mass becomes more concentrated on low-energy states, and the high-energy tail is suppressed.

4.3. Degeneracy and Model C

Model C introduces duplication penalties on certain pairs of cuts, which creates a set of degenerate ground states: in our full-adder instance there are ten assignments that achieve the same minimum classical energy. This changes both the structure of the classical energy landscape and how QAOA distributes probability.

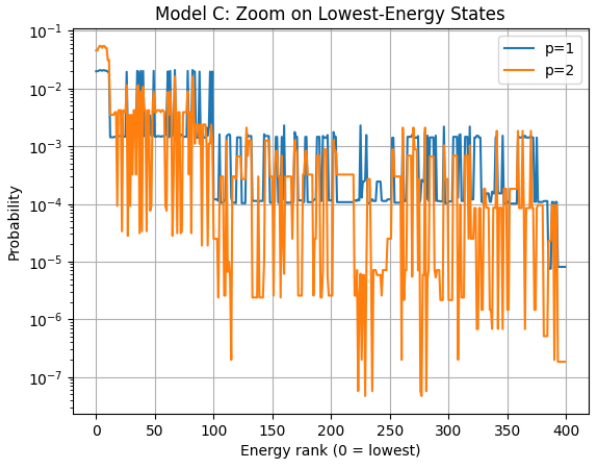

FIG. 4: Sampling probability versus classical energy rank for Model C. The flat region at the smallest ranks reflects degeneracy of the ground state manifold due to overlap penalties.

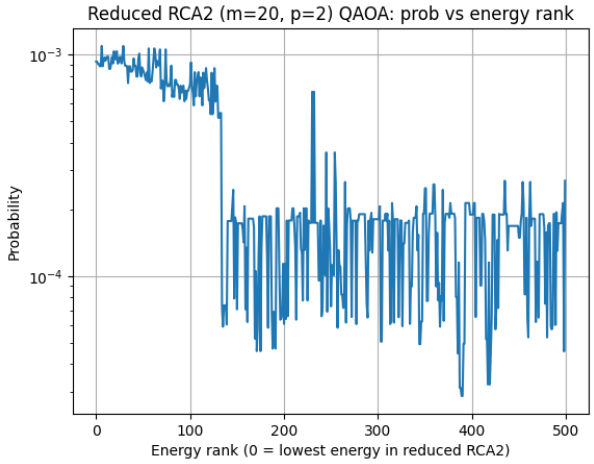

FIG. 5: Sampling probability versus classical energy rank for reduced RCA2 instances (20 variables). The trend of higher probability for lower energy states persists, but the contrast decreases as the number of variables increases.

Figure 4 shows the probability versus energy rank for Model C. The lowest-energy region forms a visible plateau: many bitstrings share the same minimum energy, and QAOA assigns comparable probability to each of them. Above this plateau, the probability still decays with increasing energy, but the curve is noticeably less smooth than in Model B. This is expected: the pairwise duplication terms in E_{dup} couple distant parts of the DAG and introduce frustrated interactions, which both increase degeneracy and make the excited-state structure more irregular. Nevertheless, QAOA continues to concentrate a significant fraction of its probability mass in the degenerate ground-state manifold.

4.4. Reduced RCA2 Experiment

The full 2-bit ripple-carry adder (RCA2) instance produces 28 QUBO variables, making full statevector simulation expensive at higher depth. Instead, we select subsets of 16–20 variables and study the corresponding reduced Hamiltonian. As shown in Figure 5, QAOA continues to bias probability toward low-energy states in these reduced instances, but the contrast between low and high energies becomes less pronounced as the dimension grows. This is consistent with the growing Hilbert-space dimension and fixed, shallow circuit depth.

5. Discussion

Across all models, QAOA exhibits qualitatively favorable behavior on these structured DAG-based problems. The energy landscape for Model B is smooth and features a clear basin of attraction, suggesting that classical optimizers can find good parameters even at low depth. The probability distributions show a clear negative correlation between energy rank and sampling probability, and increasing depth enhances the concentration of probability on low-energy states.

The contrast between Models B and C highlights how problem structure shapes QAOA behavior. In Model B, the dominant depth penalty is effectively a node-local, almost linear function of the cut choices, and violations of the exactly-one constraint are heavily suppressed by E_A. The resulting Ising Hamiltonian is close to a weighted sum of single-qubit Z_i terms plus a small number of simple couplings, which explains the smooth energy landscape and the clean, monotone probability–energy trend. Model C, by comparison, adds duplication penalties that couple nonlocal pairs of cuts; these quadratic terms create multiple degenerate optima and a more irregular spectrum of excited states. QAOA still favors low-energy configurations, but the sampling distribution reflects this richer, more frustrated structure.

For Model C, degeneracy in the ground-state man-

ifold leads to a plateau at the lowest energies, but the algorithm still preferentially samples these states. This behavior aligns with theoretical expectations for QAOA on problems with degenerate minima[3]. The reduced RCA2 experiments indicate that similar qualitative behavior persists as we move to larger DAGs, though the probability contrast weakens.

Overall, these results suggest that QAOA can exploit structure in realistic logic circuits to bias sampling toward good solutions, even when it does not concentrate all probability on the exact optimum.

6. Limitations and Future Work

The dominant bottleneck in this study is the classical outer loop: evaluating the expected energy for each set of QAOA parameters dominates runtime in statevector simulation, even for $n = 11$. On real hardware, gate noise and depth limits would further constrain feasible values of p. In addition, QUBO construction itself scales quadratically with the number of variables, and mapping realistic logic networks with hundreds or thousands of nodes would require more sophisticated preprocessing and problem decomposition.

Future work could incorporate absorption constraints and alternative cost models, explore more expressive mixer Hamiltonians adapted to the DAG structure, and investigate tensor-network or hardware-based QAOA for larger instances. A promising direction is to use QAOA as a heuristic subroutine that proposes candidate cut assignments, which are then refined or verified by classical mapping algorithms.

7. Conclusion

We have presented a QUBO formulation of FPGA cut selection and used it to study the behavior of QAOA on structured logic-DAG instances derived from a 1-bit full adder and a reduced 2-bit ripple-carry adder. By constructing Hamiltonians from real mapping problems and implementing a custom QAOA simulator, we observed smooth energy landscapes, nontrivial ground-state overlap, and meaningful probability–energy correlations at small depth.

Although QAOA in its current form is not competitive with classical mapping tools, our results provide a concrete case study of how variational quantum algorithms behave on non-random, engineering-motivated problems. As quantum hardware and hybrid methods mature, such structured benchmarks may help clarify where quantum optimization can offer practical value.

References

[1] A. Mishchenko, S. Chatterjee, and R. Brayton, in International Workshop on Logic Synthesis (IWLS) (2006) http://www-cad.eecs.berkeley.edu/~alanmi/publications/2006/fpga06_map.pdf.

[2] Z. Hu, J. Wang, H. Sun, and J. Cong, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2025), arXiv:2504.12824.

[3] E. Farhi, J. Goldstone, and S. Gutmann, arXiv:1411.4028 (2014).

[4] K. Blekos, K. Giannakis, D. Vlachos, and G. Boulougouris, arXiv:2306.09198 (2024).

[5] F. Glover, G. Kochenberger, and Y. Du, arXiv:1811.11538 (2019).

Supplementary Materials

The full Jupyter notebook used for simulation and QAOA experiments is available at:

62400_final_project.ipynb.