Application of MaxCut and QAOA in Portfolio Optimization

Oliver Adams
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, United States
adamso@rpi.edu

Batuhan Yalcin
MANE Department
Rensselaer Polytechnic Institute
Troy, United States
yalcib@rpi.edu

Levy Lin
Department of Computer Science
Department of Economics
Rensselaer Polytechnic Institute
Troy, United States
linl9@rpi.edu

Holden Mac Entee
Department of Computer Science
Department of Electrical, Computer & Systems Engineering
Rensselaer Polytechnic Institute
Troy, United States
macenh@rpi.edu

***Abstract*—The integration of quantum computing in finance offers significant potential to address the computational challenges of portfolio optimization, a task that investment firms perform continuously. Classical exact methods can be computationally expensive, while approximate solutions like the Goemans-Williamson algorithm achieve a limited approximation ratio of 87% for MaxCut. This study explores the Quantum Approximate Optimization Algorithm (QAOA) to improve upon these limitations by potentially achieving a speedup over classical deterministic methods and enhancing the approximation ratio over approximate algorithms.**

We demonstrate the functionality of our implementation on a weighted 25-complete graph. This graph considers 25 stocks and minimizes the weights between the two to create a diverse portfolio. We achieved an average fidelity of 98% using 5 repetitions on quantum hardware.

I. Introduction

The Quantum Approximate Optimization Algorithm (QAOA) [1] leverages a hybrid quantum-classical framework to solve combinatorial optimization problems. This algorithm employs a parameterized quantum circuit whose parameters are iteratively optimized by a classical optimizer. By alternating two operators k times, QAOA creates a quantum state that, when measured, is highly likely to yield an acceptable solution to the problem [2]. QAOA has the potential to outperform classical algorithms in solving combinatorial problems and NP-hard problems, supporting the vision of quantum advantage in optimization tasks.

In this study, we integrate QAOA with Modern Portfolio Theory (MPT) to address portfolio optimization, a common problem in finance. MPT provides a rigorous framework for evaluating assets by quantifying both expected return and risk, enabling the construction of portfolios that minimize risk without sacrificing returns. By representing assets as nodes in an N-complete graph, we assign edge weights based on the correlation between asset pairs. The MaxCut algorithm is then applied to partition the graph, minimizing the correlation between assets in separate groups and maximizing portfolio diversification. Additionally, MPT's ability to incorporate risk aversion levels allows for customized portfolio distributions based on individual or institutional risk preferences.

II. Related Work

A. Quantum Approximate Optimization Algorithm (QAOA)

The Quantum Approximate Optimization Algorithm (QAOA), introduced by Farhi, Goldstone, and Gutmann in 2014 [4], is a hybrid quantum-classical algorithm designed to solve combinatorial optimization problems. QAOA alternates between applying a problem-specific cost Hamiltonian and a mixing Hamiltonian, with parameters optimized in a classical loop. Initial studies demonstrated its potential for solving problems like MaxCut, showcasing quantum algorithms' ability to approximate NP-hard problems with competitive performance.

Subsequent research has focused on improving QAOA's efficiency and scalability. Huot et al. [9] explored parameter transfer strategies to enhance QAOA's performance across different problem instances, reducing computational overhead. Sureshbabu et al. [3] analyzed the algorithm's performance under varying circuit depths, highlighting trade-offs between accuracy and noise resilience in noisy intermediate-scale quantum (NISQ) devices. Despite these advancements, challenges remain in adapting QAOA to domain-specific problems and ensuring high fidelity on hardware-constrained quantum systems.

B. Modern Portfolio Theory (MPT)

Modern Portfolio Theory (MPT), introduced by Markowitz in 1952 [5], revolutionized financial decision-making by formalizing the trade-off between risk and return. By quantifying risk using the covariance matrix of asset returns, MPT enables the construction of portfolios that maximize expected returns for a given risk level or minimize risk for a desired return. This theory underpins most classical portfolio optimization techniques.

Extensions to MPT have addressed practical challenges such as incorporating transaction costs, liquidity constraints, and dynamic market conditions. Black [6] introduced the Black-Litterman model to improve portfolio allocation by combining market equilibrium data with investor views. More recently, Rockafellar and Uryasev [7] proposed using Conditional Value-at-Risk (CVaR) to optimize portfolios under tail-risk constraints. These advancements, while powerful, rely on computationally intensive optimization methods, which can be inefficient for large asset universes.

C. Portfolio Optimization Using QAOA

Integrating QAOA with portfolio optimization represents a novel approach to solving financial problems. Venturelli and Kondratyev [10] demonstrated the application of quantum annealing to portfolio optimization, highlighting its ability to efficiently find diversified portfolios.

While quantum annealing has garnered attention, studies specifically applying QAOA to portfolio optimization are limited. Huot et al. [9] proposed a hybrid quantum-classical framework using QAOA to solve portfolio optimization problems mapped to a quadratic unconstrained binary optimization (QUBO) form. Their work illustrated the feasibility of encoding financial data into a quantum framework but noted the challenges posed by hardware limitations and parameter optimization.

In addition to these efforts, Buonaiuto et al. [8] experimented with real quantum devices for portfolio optimization, demonstrating best practices for mapping financial problems to quantum systems. This work emphasized the challenges of noise and decoherence in achieving consistent results on current quantum hardware.

Our work extends these efforts by integrating QAOA with Modern Portfolio Theory using a novel edge weight to address the MaxCut problem in portfolio optimization. By introducing a parity operator to enforce problem-specific constraints, our approach achieves better fidelity and reliability in mapping stock data to a quantum Hamiltonian. Additionally, we explore the trade-offs between circuit depth and noise in achieving optimal portfolio diversification.

III. Creation of N-Complete Graph

In this section, we describe how relationships between stocks are numerically represented and how these values are mapped to a graphical framework.

A. Modern Portfolio Theory and Edge Weight Calculation

Following Modern Portfolio Theory (MPT), we construct a complete graph where each node represents an asset, and the edges represent relationships between pairs of assets. These relationships are quantified using a weighted combination of correlation, return differences, and volatility differences, normalized and adjusted based on user-defined parameters. This approach ensures a comprehensive representation of the trade-offs between diversification, return maximization, and risk minimization.

The edge weight between two assets, i and j, is calculated using the following formula:

$$\begin{aligned} \text{Edge Weight}_{i,j} &= \alpha \cdot (1 - \text{Correlation}_{i,j}) \\ &+ \beta \cdot \text{Normalized Return Difference}_{i,j} \\ &+ \gamma \cdot \text{Normalized Volatility Difference}_{i,j}, \end{aligned} \tag{1}$$

where α, β, and γ are user-defined weights for the respective components.

1) Correlation Component: The correlation component quantifies the linear relationship between the historical returns of two assets, i and j, using the Pearson correlation coefficient:

$$\text{Correlation}_{i,j} = \frac{\text{Cov}(R_i, R_j)}{\sigma_{R_i} \cdot \sigma_{R_j}}, \tag{2}$$

where $\text{Cov}(R_i, R_j)$ is the covariance of the returns, and σ_{R_i} and σ_{R_j} are the standard deviations of the returns for assets i and j, respectively.

2) Normalized Return Difference: The return difference quantifies the disparity in expected returns between two assets. It is normalized to ensure comparability across all asset pairs:

$$\begin{aligned} &\text{Return Difference}_{i,j} = \\ &\left|\text{Expected Return}_i - \text{Expected Return}_j\right|, \end{aligned} \tag{3}$$

$$\begin{aligned} &\text{Normalized Return Difference}_{i,j} = \\ &\frac{\text{Return Difference}_{i,j}}{\max(\text{Return Differences})}. \end{aligned} \tag{4}$$

3) Normalized Volatility Difference: The volatility difference measures the absolute difference in historical volatility between two assets. Like the return difference, it is normalized for consistency:

$$\begin{aligned} &\text{Volatility Difference}_{i,j} = \\ &\left|\text{Volatility}_i - \text{Volatility}_j\right|, \end{aligned} \tag{5}$$

$$\begin{aligned} &\text{Normalized Volatility Difference}_{i,j} = \\ &\frac{\text{Volatility Difference}_{i,j}}{\max(\text{Volatility Differences})}. \end{aligned} \tag{6}$$

B. Data Acquisition and Processing

The data required to compute the edge weights was obtained from historical stock data for the S&P 500. Using the `yFinance` library in Python, we retrieved and stored historical adjusted closing prices for each asset over 2 years from Yahoo Finance. From this data, we computed the average price, variance, standard deviation, and expected returns for each asset. Using the aforementioned formulas, these metrics were then used to calculate the edge weights.

By combining correlation, return differences, and volatility differences, the constructed graph ensures that the resulting MaxCut solution promotes portfolio diversification while aligning with the principles of MPT. The incorporation of user-defined weights α, β, and γ allows for flexibility in adjusting the optimization based on specific risk and return preferences or employing computational optimization to those weights.

C. Mapping to Nodes

The portfolio was represented as an *N*-complete graph, where each of the *N* assets corresponds to a node, and every pair of nodes is connected by an undirected, weighted edge. This graph representation enables intuitive visualization of the relationships among assets and facilitates the application of graph-based optimization algorithms.

The MaxCut algorithm is employed to partition the graph into two complementary sets, *S* and *T*, such that the sum of the edge weights (representing correlations) between the two sets is maximized. In the context of portfolio optimization, this partitioning aims to minimize risk and maximize returns. Specifically:

- **Risk Minimization:** By separating highly correlated assets, the resulting subsets reduce the likelihood of simultaneous adverse movements.
- **Return Maximization:** Ensuring diversification across subsets aligns with Modern Portfolio Theory (MPT), which seeks an optimal trade-off between risk and return.

The incorporation of the desired risk aversion ratio, derived from MPT, further refines the optimization by balancing the portfolio's overall risk and return characteristics. This approach translates a traditionally numerical problem into a graph-theoretic framework, solvable using advanced quantum algorithms such as the Quantum Approximate Optimization Algorithm (QAOA).

IV. Circuit Construction and Parameter Trends

In this section, we discuss how MaxCut was mapped to a Hamiltonian operator and implemented in the Quantum Approximate Optimization Algorithm (QAOA). We also describe the classical optimizer used in both simulation and experimental implementations.

A. Cost and Mixer Layer Construction

To apply MaxCut on a quantum circuit, the cost function for MaxCut is first defined as:

$$C(x) = \sum_{i,j=1}^{n} W_{ij} x_i (1 - x_j), \tag{7}$$

where W_{ij} represents the edge weights, and x_i and x_j are binary variables denoting the cut.

To incorporate this cost function into QAOA, we define a Hamiltonian H_C such that:

$$H_C \left| x \right\rangle = C(x) \left| x \right\rangle . \tag{8}$$

The classical variables are expressed in terms of their quantum mechanical equivalents:

$$W_{ij} = -Q_{ij}, \tag{9}$$

$$x_i = \frac{1 - Z_i}{2}, \tag{10}$$

$$x_j = \frac{1 - Z_j}{2}, \tag{11}$$

where Z_i and Z_j are the Pauli-Z operators acting on the qubits corresponding to nodes i and j.

Substituting these into the cost function, the Hamiltonian for MaxCut is given by:

$$H_C = \sum_{i,j=1}^{n} Q_{ij} Z_i Z_j + \sum_{i,j=1}^{n} Q_{ij} Z_i. \tag{12}$$

This Hamiltonian is applied in the cost layer of QAOA, defined as:

$$U_C(\gamma_i) = e^{-i\gamma_i H_C}, \tag{13}$$

and paired with a mixer layer:

$$U_M(\beta_i) = e^{-i\beta_i H_M}, \tag{14}$$

where $H_M = \sum_{i=1}^{n} X_i$ is the mixing Hamiltonian, and X_i is the Pauli-X operator acting on qubit i.

The final gate-level implementations for the cost and mixer layers are:

$$U_C(\gamma_i) = \prod_{i,j=1}^{n} R_{Z_i Z_j}(Q_{ij}\gamma), \tag{15}$$

$$U_M(\beta_i) = \prod_{i=1}^{n} R_x(2\beta). \tag{16}$$

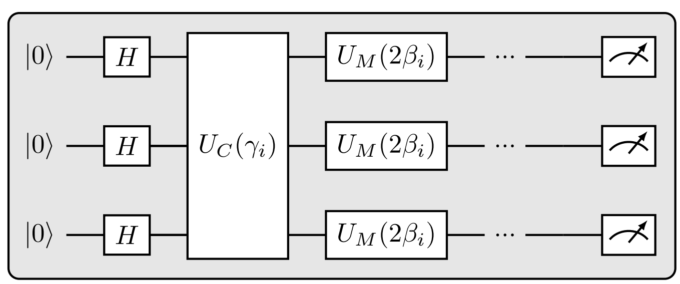

Fig. 1. Full circuit diagram for QAOA.

Figure 1 illustrates the circuit diagram for QAOA, showing the alternating cost and mixer layers, parameterized by γ_i and β_i, respectively. These layers are repeated k times, with increasing circuit depth improving the probability of finding the optimal solution. However, deeper circuits are more susceptible to decoherence and noise, requiring a balance between depth and accuracy.

B. Incorporating a Parity Operator

To enforce constraints on the MaxCut problem, a parity operator was added to the Hamiltonian. This operator penalizes states that violate the constraints, ensuring valid solutions.

Mathematically, the parity operator penalizes individual node states as:

$$H_P = \sum_{i=1}^{n} P_i Z_i, \tag{17}$$

where P_i is the penalty weight for node i. This operator ensures that the resulting solutions are both valid and optimal with respect to the MaxCut constraints.

The parity operator was added to the cost Hamiltonian as:

$$H_{\text{total}} = H_C + H_P, \tag{18}$$

where H_P enforces the parity constraints. This modification improves the reliability of the quantum solution, aligning it with the problem requirements.

C. Parameter Optimization

Parameter optimization is a critical step in the implementation of QAOA, as the performance of the algorithm heavily depends on how effectively the parameters are tuned. In this work, we use the COBYLA classical optimizer to optimize the circuit parameters. COBYLA is well-suited for problems with continuous parameters and no explicit gradients, making it a practical choice for QAOA on NISQ devices.

In addition to optimizing the circuit parameters, we also focused on optimizing the weights α, β, and γ used in the edge weight calculation for the MaxCut graph. These parameters were tuned to maximize the risk-adjusted returns of the portfolio generated by the resulting MaxCut solution. The optimization process involved systematically evaluating combinations of α, β, and γ across the solution space and identifying the configuration that yielded the highest Sharpe ratio for the portfolio.

Presently, our implementation begins by solving MaxCut on an unweighted graph of the same size. The parameters optimized for the unweighted graph are then used as initial values for the weighted graph optimization, which is also performed using COBYLA. This structured approach improves the efficiency of the optimization process compared to random initialization.

While we have not implemented machine learning (ML)-based parameter prediction [1] in this work, we acknowledge its potential to further accelerate parameter optimization. Future work may explore training ML models to predict near-optimal parameters for high-depth QAOA circuits based on data from lower-depth circuits. Such methods could significantly reduce the computational cost of the classical optimizer loop and enhance the scalability of QAOA implementations.

V. Results

A. Quantum Algorithm Performance

To evaluate the effectiveness of the QAOA implementation, we analyzed the expected MaxCut value as a function of the number of shots for the 25-node graph. Figure 2 shows the relationship, highlighting convergence trends and the impact of circuit depth on solution quality. Our results demonstrate that with sufficient iterations, the quantum algorithm approaches the theoretical MaxCut value, achieving a fidelity of 98%.

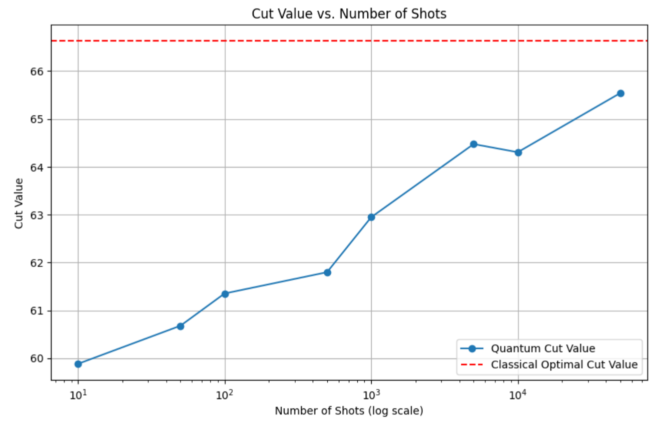

Fig. 2. Comparison of results from 25-node graph over the number of shots.

Furthermore, we compared the approximation quality of QAOA to classical methods. Table I summarizes the achieved MaxCut values for graphs of varying sizes. Despite the constraints of current quantum hardware, QAOA solutions showed promising scalability.

TABLE I
FIDELITY OF QUANTUM HARDWARE QAOA COMPARED TO CLASSICAL MAXCUT SOLUTIONS

Num Nodes	Classical MaxCut	QAOA MaxCut	Fidelity (%)
5	3.677089	3.677089	100%
10	13.80871	13.80871	100%
15	26.36673	26.05348	98.81%
20	42.75414	42.37184	99.10%
25	66.62953	65.90017	98.90%

B. Computational Efficiency

A comparison of execution times for quantum and classical methods is shown in Figure 2. Classical solvers outperformed quantum methods for small graphs ($N \leq 20$), but quantum methods showed potential for scalability as N increased. The inclusion of a parity operator introduced minimal overhead while significantly improving constraint enforcement. We also acknowledge that our classical computation was run on a single-core where, as distributed systems would provide a large speedup, but we still believe this to be a fair comparison.

C. Portfolio Quality

To assess the financial impact of our approach, we evaluated the portfolio's Sharpe ratio before and after applying the MaxCut solution. As shown in Table II, the post-cut portfolio

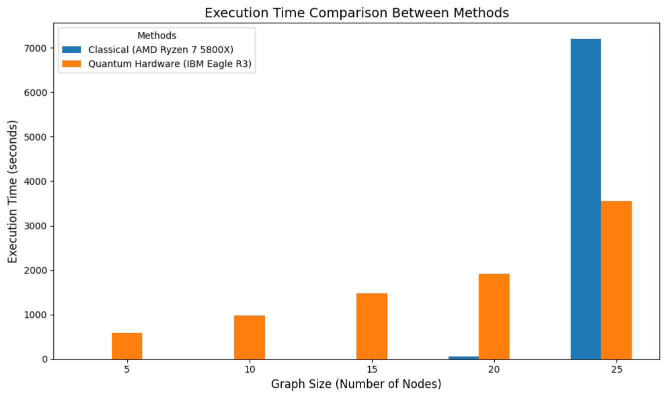

Fig. 3. Comparison of execution times on classical hardware and quantum hardware.

achieved an average 140% improvement in the Sharpe ratio, driven by reduced correlations and improved diversification.

TABLE II
EVALUATING THE SHARPE RATIOS OF THE GRAPHS PRE AND POST MAXCUT

Num Nodes	Pre-MaxCut	Post-MaxCut	Improvement (%)
5	17.87472	21.05055	117%
10	16.27895	22.55111	138%
15	18.09689	34.55607	190%
20	19.07402	23.19470	121%
25	16.58032	22.77069	137%

Overall, our results highlight the potential of QAOA in financial portfolio optimization, with clear improvements in portfolio quality and a foundation for further scalability in quantum methods.

VI. CONCLUSION

In this work, we have demonstrated how the Quantum Approximate Optimization Algorithm (QAOA) can be integrated with Modern Portfolio Theory (MPT) to tackle the portfolio optimization problem. By formulating the problem as a MaxCut instance on an N-complete graph and carefully mapping it to a corresponding Hamiltonian, we leveraged QAOA to identify asset groupings that enhance diversification and improve risk-adjusted returns. Our results on a weighted 25-complete graph underscore the potential of quantum algorithms to approximate NP-hard optimization problems, achieving a fidelity of nearly 99% when compared to classical solutions. Furthermore, the incorporation of a parity operator into the Hamiltonian improved the consistency and quality of the results by enforcing problem-specific constraints, and the subsequent improvement in Sharpe ratios illustrated the practical financial benefits of this approach.

However, current quantum hardware limitations, such as constrained qubit counts, limited coherence times, and gate execution fidelity, impose challenges on the scalability and precision of QAOA in real-world financial settings. Despite these constraints, our study highlights a promising pathway towards achieving quantum advantage in portfolio optimization, suggesting that as quantum hardware and noise mitigation techniques improve, so too will the scope and accuracy of QAOA-driven solutions.

A. Future Work

Several directions exist for extending this research and overcoming current limitations:

- **Hardware and Noise Mitigation:** As quantum devices evolve, exploiting error mitigation techniques, advanced calibration protocols, and improved qubit connectivity could enable deeper QAOA circuits and more accurate solutions for larger problem instances.
- **Parameter Optimization and Machine Learning:** Incorporating machine learning techniques to predict near-optimal parameters for QAOA may dramatically reduce the computational burden of parameter tuning. This approach could involve training models on smaller problem instances to inform parameter choices for larger, more complex graphs.
- **Dynamic Market Conditions:** Future investigations might incorporate time-varying models of returns and correlations, enabling the QAOA framework to adapt portfolios dynamically in response to shifting market conditions.
- **Alternative Objective Functions and Constraints:** While this study focused on traditional risk and return metrics, exploring more sophisticated constraints—such as liquidity, transaction costs, or regulatory requirements—could enhance the real-world applicability of QAOA-based portfolio optimization.

By addressing these challenges and pursuing these avenues of research, we anticipate that QAOA, supported by continuous hardware and algorithmic advancements, will emerge as a valuable tool for complex financial optimizations and strategic decision-making in the era of quantum computing.

REFERENCES

[1] M. Alam, A. Ash-Saki, and S. Ghosh, "Accelerating Quantum Approximate Optimization Algorithm using Machine Learning," *arXiv preprint arXiv:2002.01089*, 2020.
[2] R. Shaydulin, et al., "Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem," *Science Advances*, vol. 10, no. 22, p. eadm6761, 2024.
[3] S. H. Sureshbabu, et al., "Parameter Setting in Quantum Approximate Optimization of Weighted Problems," *arXiv preprint arXiv:2305-15201*, 2024.
[4] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
[5] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
[6] F. Black, "Capital Market Equilibrium with Restricted Borrowing," *The Journal of Business*, vol. 45, pp. 444–455, 1972.
[7] R. T. Rockafellar and S. Uryasev, "Optimization of Conditional Value-at-Risk," *Journal of Risk*, vol. 2, no. 3, pp. 21–41, 2000.
[8] G. Buonaiuto, F. Gargiulo, G. De Pietro, et al., "Best practices for portfolio optimization by quantum computing, experimented on real quantum devices," *Scientific Reports*, vol. 13, p. 19434, 2023. DOI: 10.1038/s41598-023-45392-w.

[9] C. Huot, K. Kea, T. -K. Kim and Y. Han, ”Enhancing Knapsack-Based Financial Portfolio Optimization Using Quantum Approximate Optimization Algorithm,” in *IEEE Access*, vol. 12, pp. 183779-183791, 2024, doi: 10.1109/ACCESS.2024.3506981.
[10] D. Venturelli, A. Kondratyev, ”Reverse quantum annealing approach to portfolio optimization problems,” *arXiv preprint arXiv:1810.08584*, 2019.