Quantum-Enhanced Portfolio Optimization

A Hybrid Factor-QAOA Approach

Saatwik Chatkara

Caleb Fianku Quao

WISER

Project Summary

- Our project focuses on creating a robust hybrid quatum-classical framework:

1. how to efficiently solve high-dimensional optimization problem to help in ETF creations and index tracking.

2. how to address challenges in portfolio optimization, especially the noisy and unstable estimation of asset return covariance matrices.

Within the project, the quantum-enhanced optimized portfolio's performance is compared against a classical minimum variance portfolio benchmark : GUROBI on the metrics of annualized returns, Sharpe Ratio, and risk (standard deviation).

The project demonstrates the feasibility of applying hybrid quantum algorithms to complex financial challenges and their potential to uncover more efficient and diversified portfolios that perform well in the real-world scenarios.

Key Features

1. **Robust Factor Model via PCA**

- Identifies latent, uncorrelated risk factors for stable covariance matrices, reducing the impact of noise and estimation errors.

2. **Classical Portfolio Optimization Benchmark**

- Provides a quantifiable baseline for comparison to evaluate quantum advantage.

3. **Precise QUBO Problem Formulation**

- Maps financial problems under the balancing risk minimization, return maximization, and strict adherence to a cardinality constraint to a QUBO matrix.
- 4. **PennyLane for QAOA Core**
- Utilizes PennyLane for constructing, executing, and optimizing the QAOA circuit, abstracting low-level quantum gate operations.

5. **Systematic Hyperparameter Tuning**

- Implements a robust grid search to explore multi-dimensional hyperparameter space, including risk aversion, penalty strength, and QAOA layers.

Wiser Quantum Project

Factor Model for Covariance Estimation

- We implement a statistical factor model using Principal Component Analysis (PCA). The underlying assumption is that a substantial portion of asset return variance can be explained by a few common factors, with the remaining variance attributed to idiosyncratic (asset-specific) risk.
- The robust covariance matrix is reconstructed using the factor model equation: $\Sigma = B\Sigma_f B^T + D$

where Σ_f = Factor Covariance matrix

B = Factor Loadings vector

D = Diagonal Idiosyncratic Covariance

Portfolio Optimization as a QUBO Problem

- Our problem aims to minimize risk, maximize return, and enforce cardinality constraints, combined into a single cost function.

$$C(\mathbf{x}) = \mathbf{x}^T \Sigma \mathbf{x} - q \cdot \boldsymbol{\mu}^T \mathbf{x} + \lambda(\sum_{i=1}^{N} x_i - K)^2$$

- To leverage quantum computing, portfolio selection is mapped into a Quadratic Unconstrained Binary Optimization (QUBO) format.

Minimize $C(x) = x^T Q x$ where x is a vector of binary decision variables ($x_i \in \{0,1\}$, 1 if asset i is selected.
QUBO matrix Q is constructed as:

- $Qii = \Sigma ii - q \cdot \mu i + \lambda(1 - 2K)$
- $Qij = \Sigma ij + 2\lambda$ (for $i \neq j$, contributing to $Qijxixj + Qjixjxi$, where Q is symmetric)

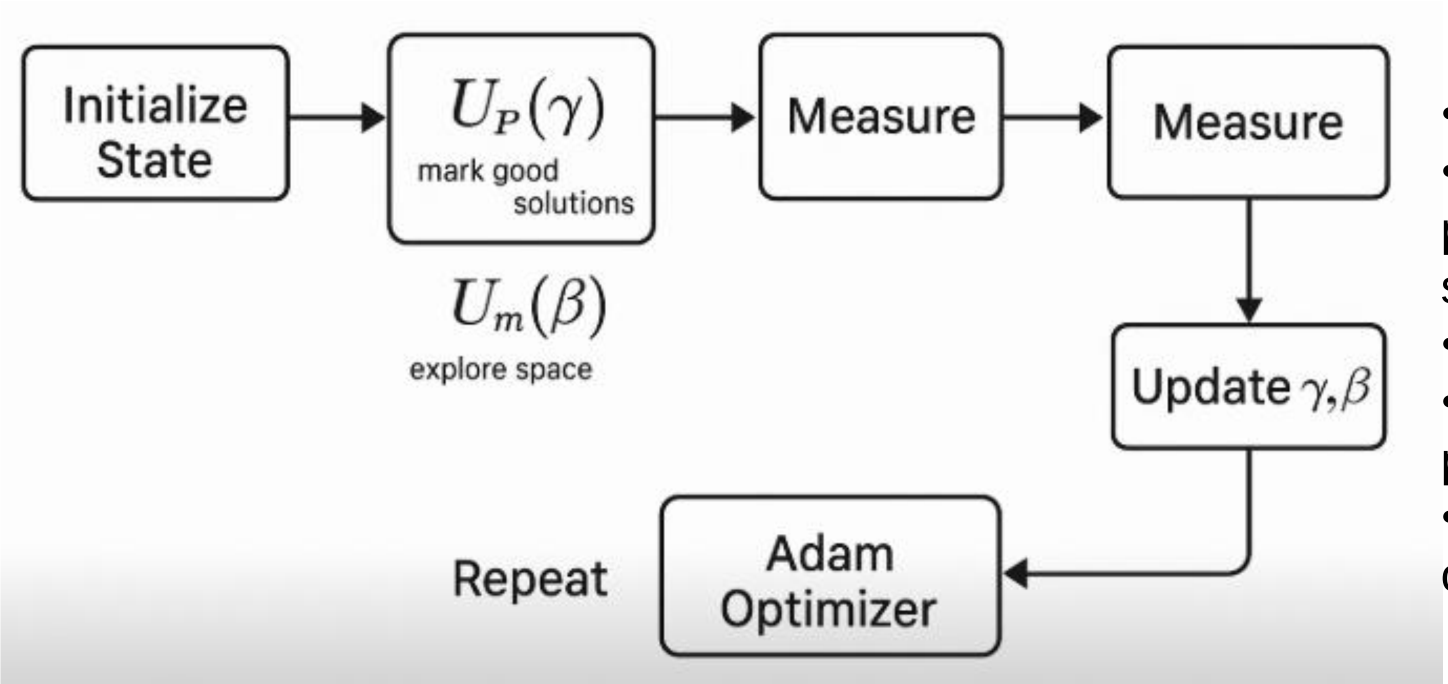
Wiser Quantum Project

QAOA Implementation with PennyLane

- QAOA is a powerful variational quantum algorithm, operating by adjusting parameters of a quantum circuit, which then generates a quantum state from which solutions can be sampled.
- **Quantum Device Initialization**
- We use `lightning.qubit` simulator.
- Using "wires = num_qubits": Each qubit represents an asset (xi), so num_qubits = num_assets
- Using shots = 10000 : crucial parameter for qml.sample measurements which is essential to obtain statistically meaningful distributions.

- We thereby construct the Cost Hamiltonian and the Mixer Hamiltonian which are required for the QAOA solver.

QAOA workflow



•Initialization of the quantum state.
•Alternating Problem Unitary UP(γ)) to encode portfolio cost and Mixer Unitary UM(β)) to explore solutions.
•Measurement of the cost (objective function).
•Classical optimization (Adam) to update parameters γ,β
•Repetition of the loop until convergence to the optimal portfolio.

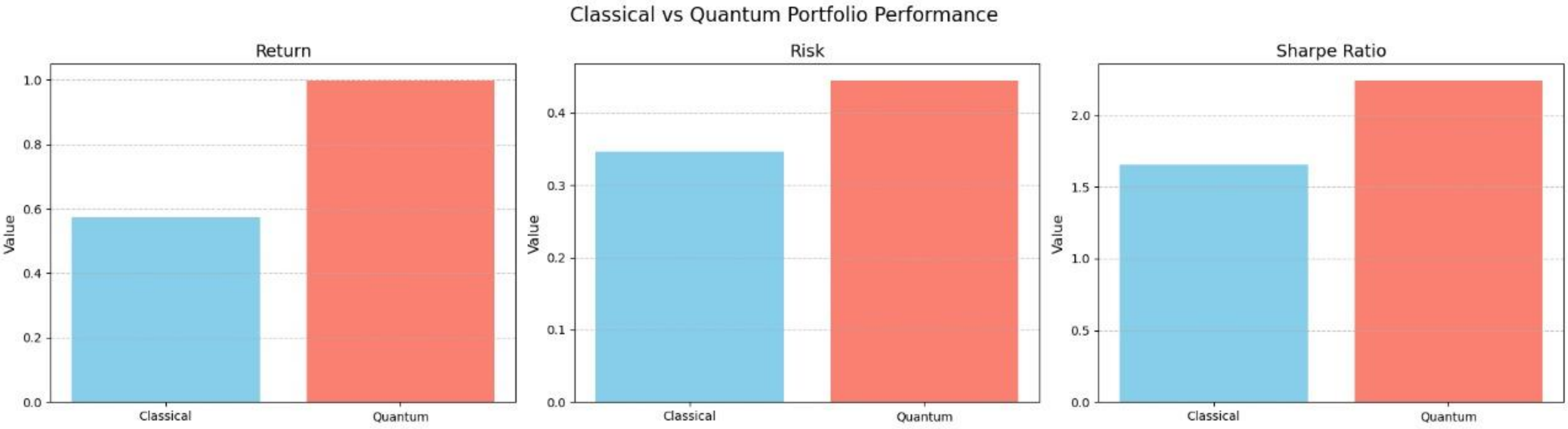
Wiser Quantum Project

Hyperparameter Tuning

- QAOA's performance hinges on its hyperparameters, which are external to the quantum circuit's variational parameters and are tuned classically through an exhaustive grid search.
- The tuning process iterates through every hyperparameter combination, involving these crucial steps:

1. QUBO Construction : rebuilding the QUBO problem for each combination
2. Multiple QAOA Runs : common practice to mitigate the problem of local minima in variational algorithms.
3. Best Parameter Selection : Identify optimal γ & β angles
4. Solution Evaluation : evaluation of valid bitstrings that satisfy constraints
5. Best Portfolio : the bitstring for specific hyperparameter combination with highest Sharpe Ratio is identified

QAOA portfolio optimization results



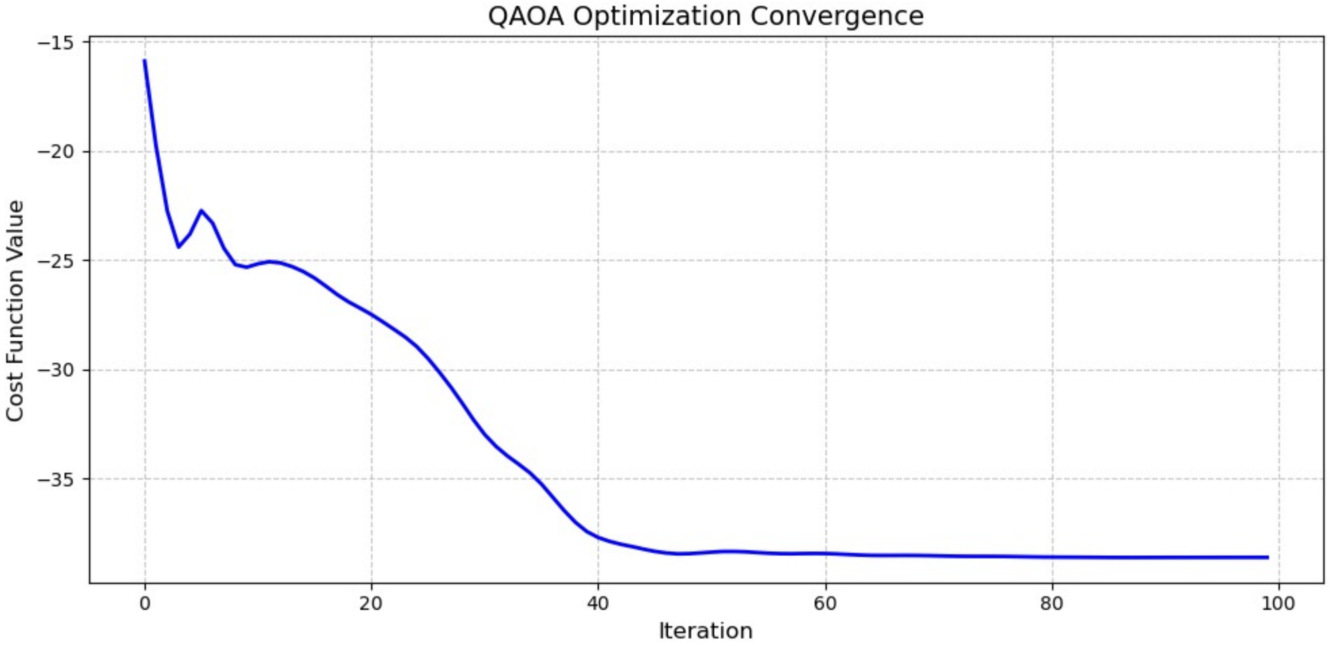
Wiser Quantum Project

QAOA portfolio optimization results

Cost function steadily minimized, and stabilizes around iteration 50, showing the successful parameter optimization.

Overall Best QAOA Portfolio

- Found via exhaustive hyperparameter search.
- Achieved highest Sharpe Ratio across all configurations.



Future Enhancements

- This project lays a robust foundation for quantum-enhanced portfolio optimization, with several exciting avenues for future work.

1. **Advanced Hyperparameter Optimization**

- Adaptive and Dynamic Tuning
- Beyond Grid-search : RL, Bayesian Optimization, Random Search, etc.

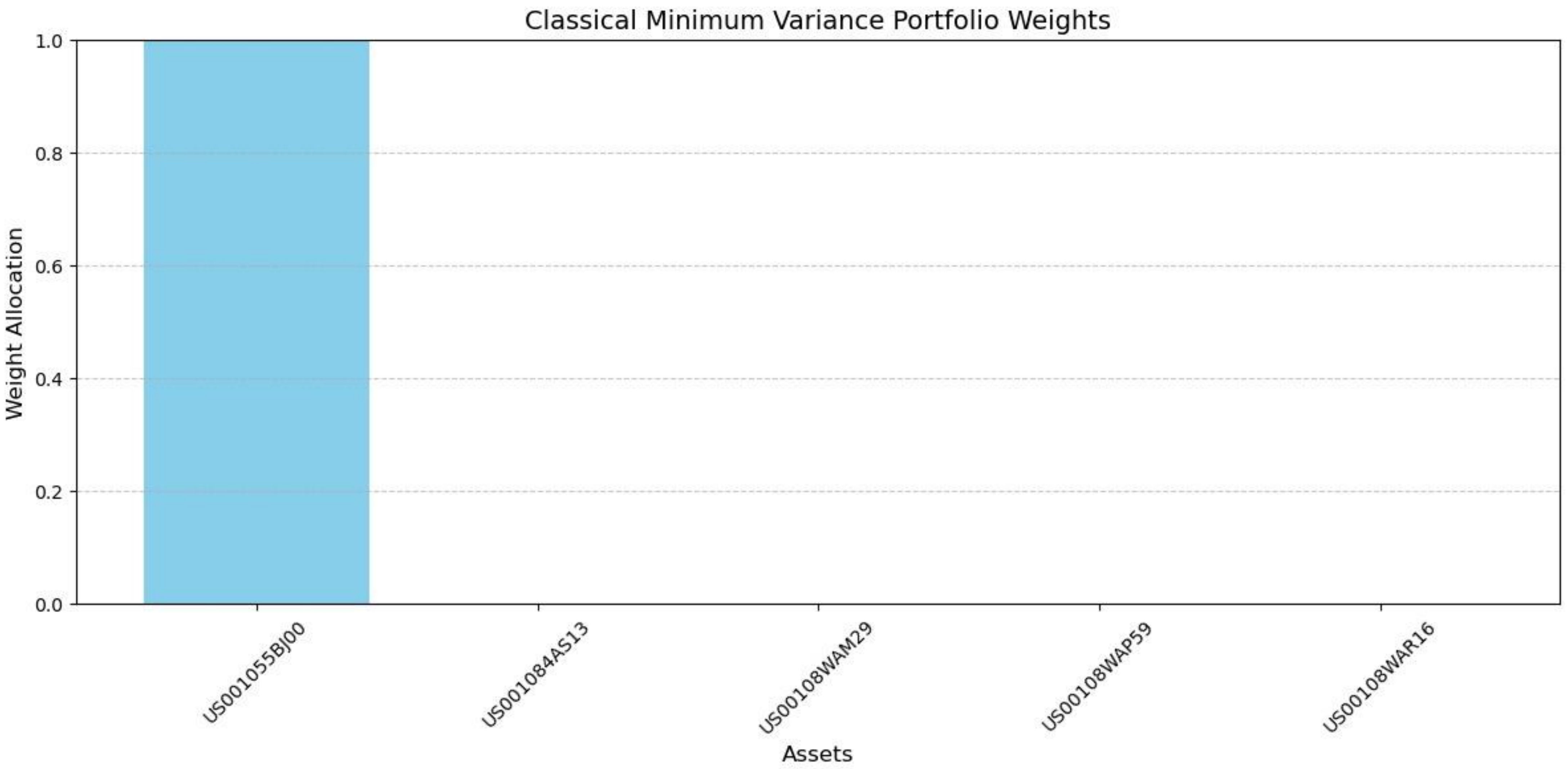
2. **Exploring Sophisticated QAOA Ansätze**

- Experiment with custom mixer Hamiltonians that may lead to faster convergence.
- Higher Layer Depths

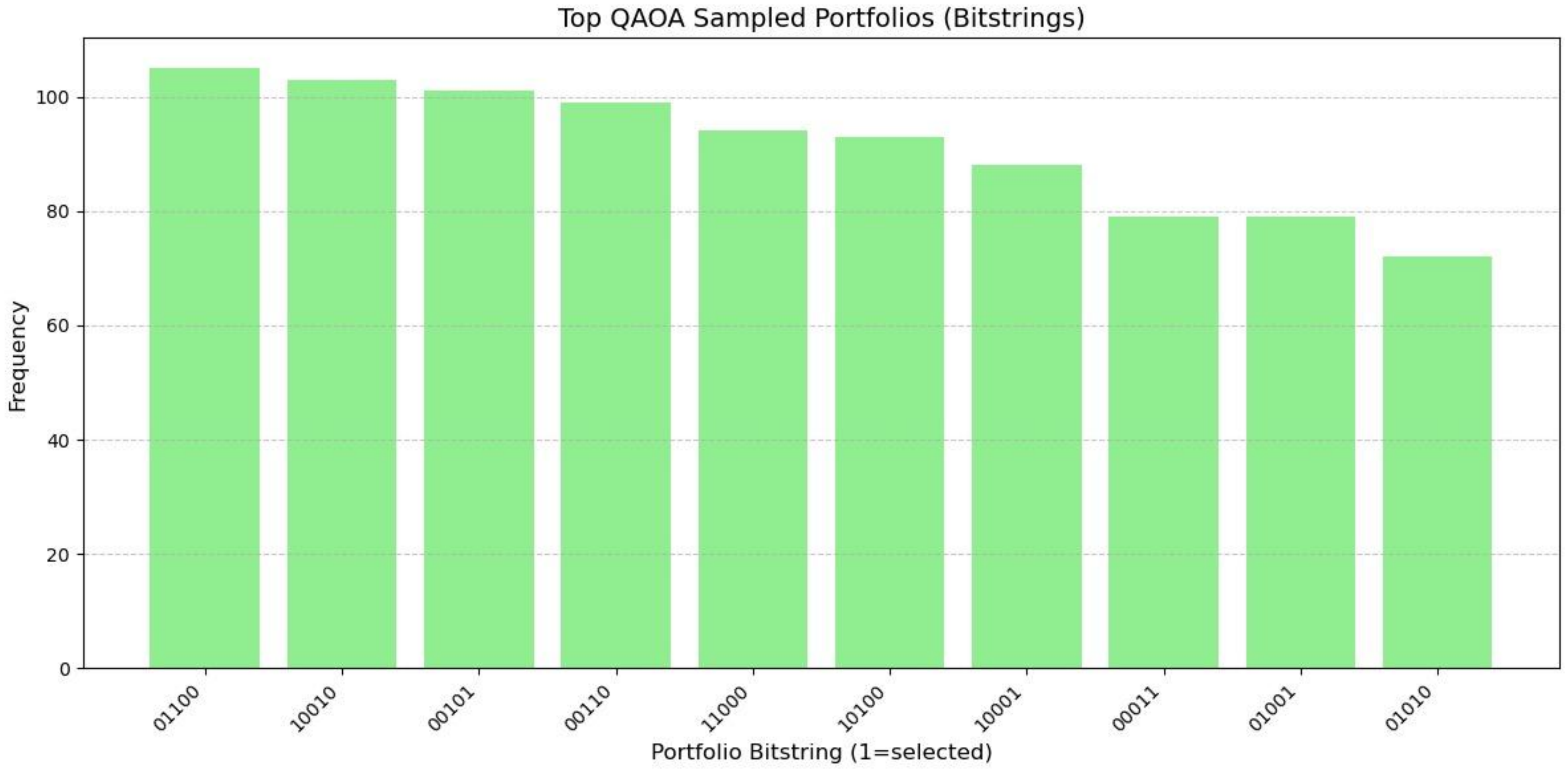
3. **Integration with other Quantum Algorithms**

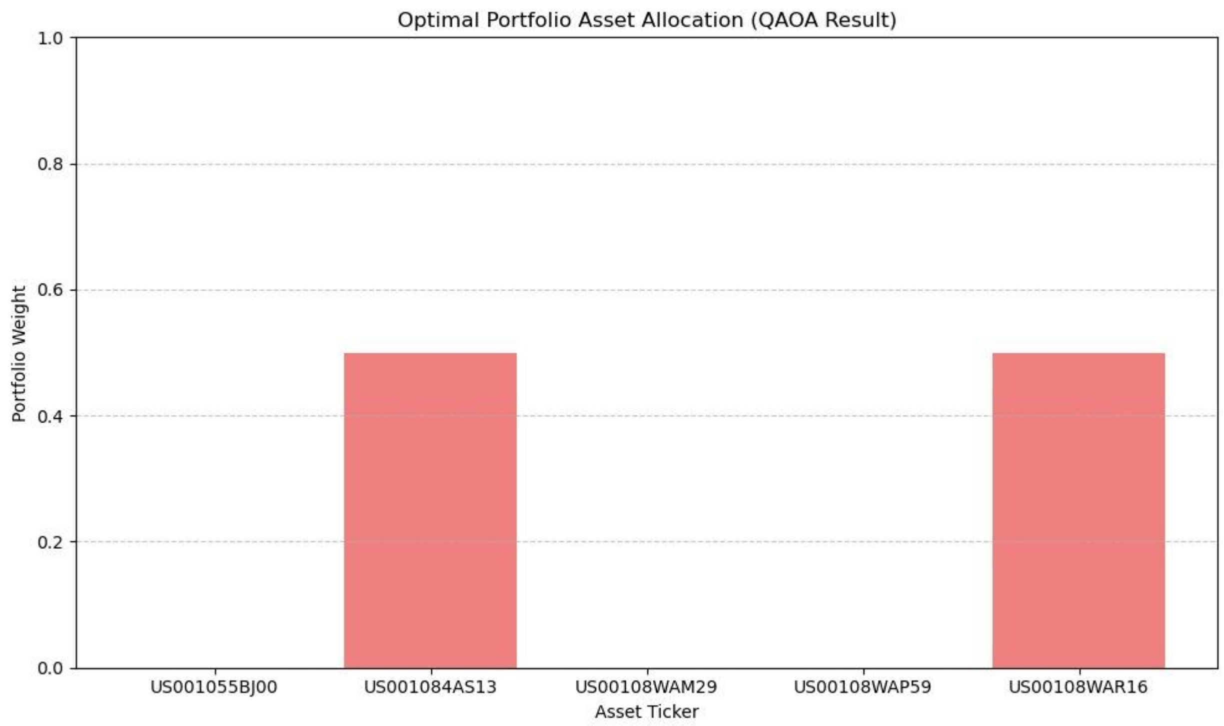
- Integration with VQE, Quantum Annealing, or Grover's Algorithm

Wiser Quantum Project

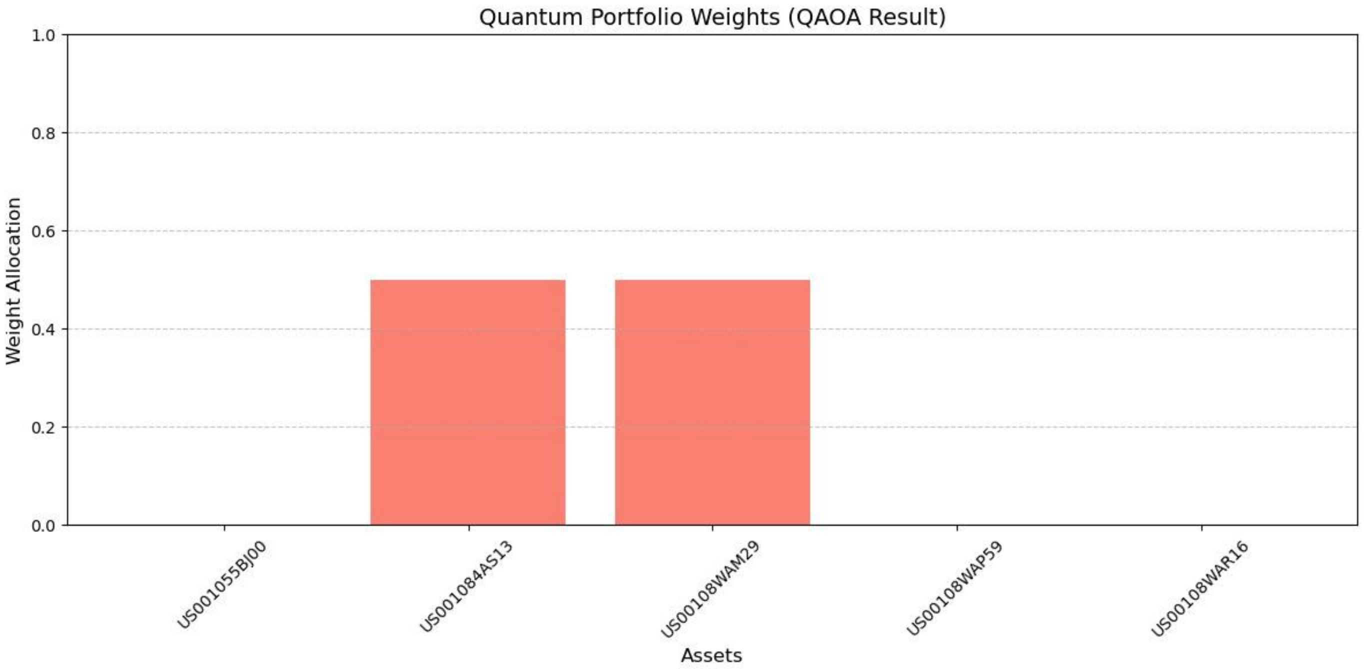


Classical Minimum Variance Portfolio Weights
Weight Allocation
1.0
0.8
0.6
0.4
0.2
0.0
US001055BJ00
US001084AS13
US00108WAM29
US00108WAP59
US00108WAR16
Assets

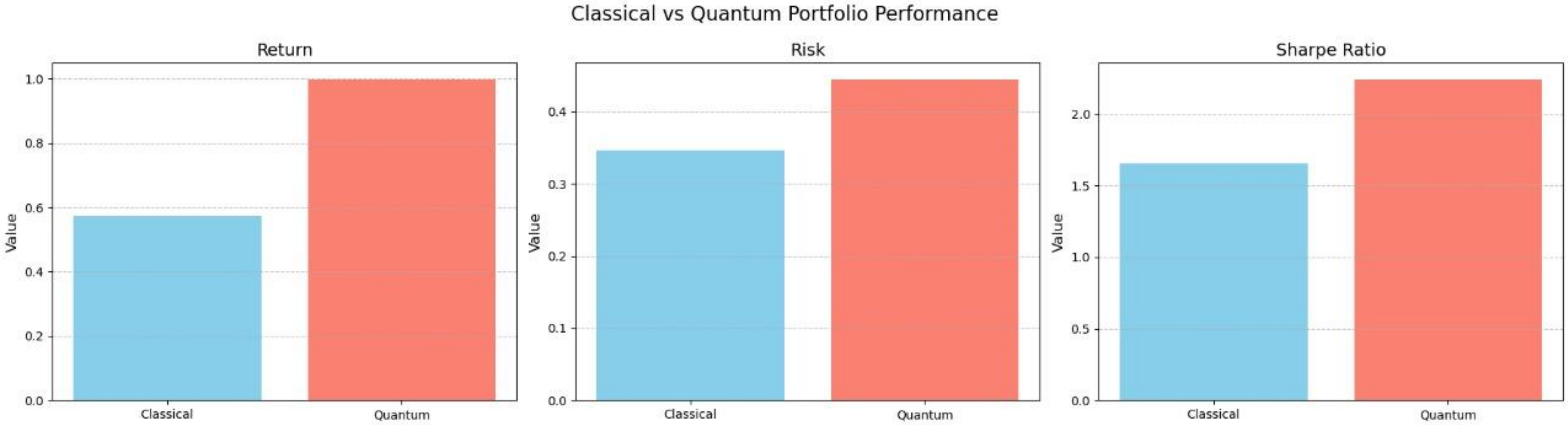

Top QAOA Sampled Portfolios (Bitstrings)
Frequency
0
20
40
60
80
100
01100
10010
00101
00110
11000
10100
10001
00011
01001
01010
Portfolio Bitstring (1=selected)

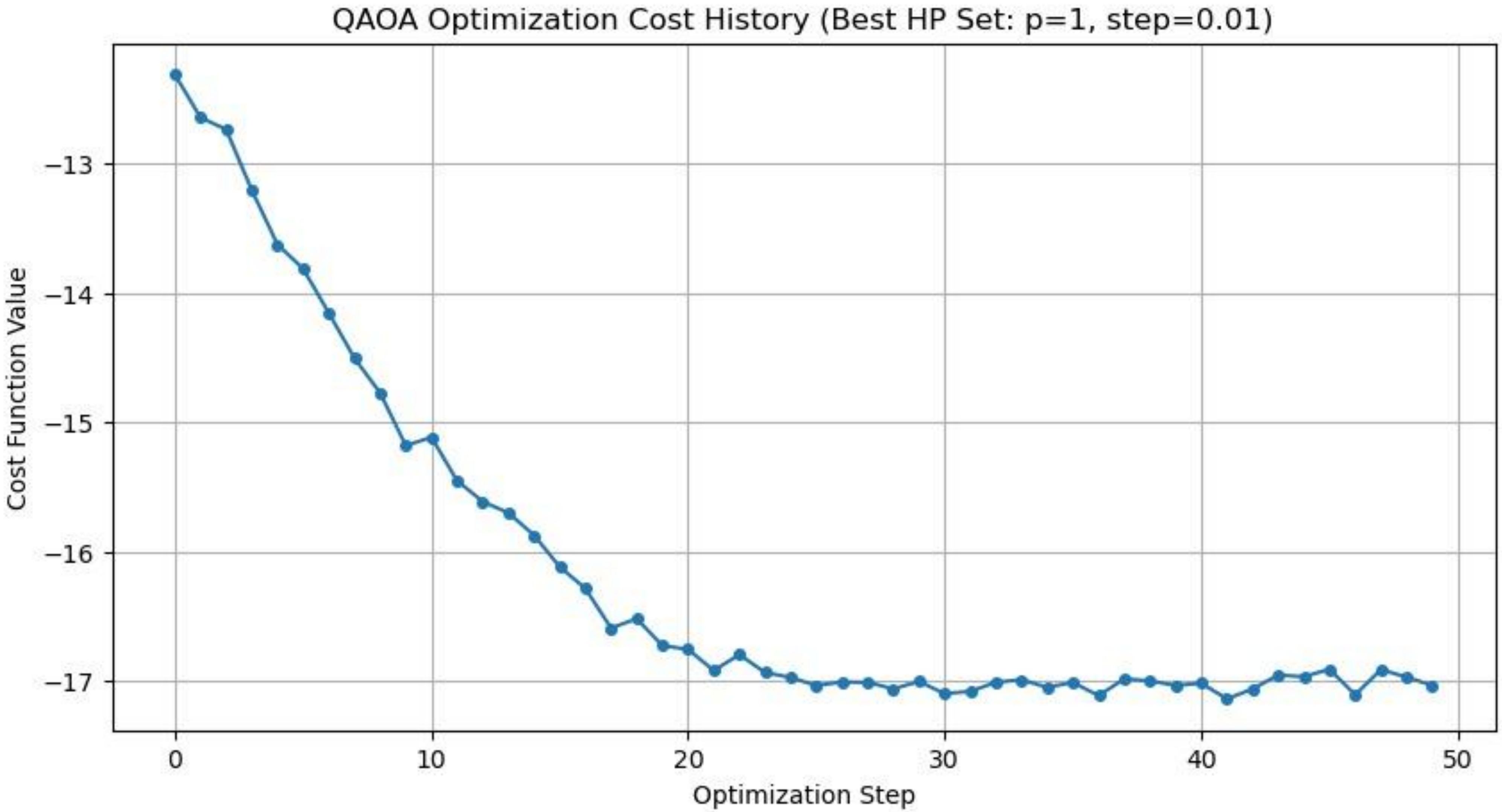


Optimal Portfolio Asset Allocation (QAOA Result)
Portfolio Weight
1.0
0.8
0.6
0.4
0.2
0.0
US001055BJ00
US001084AS13
US00108WAM29
US00108WAP59
US00108WAR16
Asset Ticker



Quantum Portfolio Weights (QAOA Result)
Weight Allocation
1.0
0.8
0.6
0.4
0.2
0.0
US001055BJ00
US001084AS13
US00108WAM29
US00108WAP59
US00108WAR16
Assets


Classical vs Quantum Portfolio Performance
Return
Risk
Sharpe Ratio
Value
1.0
0.8
0.6
0.4
0.2
0.0
0.4
0.3
0.2
0.1
2.0
1.5
1.0
0.5
Classical
Quantum



QAOA Optimization Cost History (Best HP Set: p=1, step=0.01)
Cost Function Value
−13
−14
−15
−16
−17
0
10
20
30
40
50
Optimization Step

```
--- Overall Best QAOA Portfolio from Tuning ---
Optimal Hyperparameters: q_risk_aversion=0.1, lambda_penalty=5.0, p_layers=1, stepsize=0.01
Best QAOA Portfolio (by Sharpe Ratio) Bitstring: 01001
Selected Assets: ['US001084AS13', 'US00108WAR16']
Number of selected assets: 2

======================================================================

QUBO Matrix (Q) for Overall Best QAOA Portfolio:
              US001055BJ00  US001084AS13  US00108WAM29  US00108WAP59  \
US001055BJ00    -14.937015     10.177864     10.129394     10.151953
US001084AS13     10.177864    -14.862626     10.192775     10.226383
US00108WAM29     10.129394     10.192775    -14.929957     10.164691
US00108WAP59     10.151953     10.226383     10.164691    -14.891143
US00108WAR16     10.165380     10.246386     10.179243     10.210493

              US00108WAR16
US001055BJ00     10.165380
US001084AS13     10.246386
US00108WAM29     10.179243
US00108WAP59     10.210493
US00108WAR16    -14.880095

Cost Hamiltonian (H_cost) for Overall Best QAOA Portfolio:
-11.789276712471644 * I([0, 1, 2, 3, 4]) + -2.687640556951498 * Z(0) + -2.7795395029301155 * Z(1) + 2.544466122107877 * (Z
(0) @ Z(1)) + -2.7015475358554144 * Z(2) + 2.5323485950674423 * (Z(0) @ Z(2)) + -2.7428087843127735 * Z(3) + 2.537988285147
35 * (Z(0) @ Z(3)) + -2.7603278570831256 * Z(4) + 2.541344900137121 * (Z(0) @ Z(4)) + 2.5481937069481377 * (Z(1) @ Z(2)) +
2.556595851474747 * (Z(1) @ Z(3)) + 2.5615966163864057 * (Z(1) @ Z(4)) + 2.5411728343976585 * (Z(2) @ Z(3)) + 2.54481083365
38624 * (Z(2) @ Z(4)) + 2.5526232042839694 * (Z(3) @ Z(4))

======================================================================
```