Markowitz Portfolio Optimization with quantum computing

THOMASSIN Pablo, COZ Olivier, BERDOUS Louiza
M2MO - Université Paris Cité

April 17, 2025

Overview

1. Context & Literature review
2. Theoretical Results
3. Implementation & Results
4. Replication of the results
5. Conclusion

Context & Literature review

Summary

- **Quantum Computing Revolution**
  - Promising for combinatorial optimization, especially in finance.
  - Key players: Google, IBM, D-Wave, IonQ.
- **Portfolio Optimization Problem**
  - Classic mean-variance model (Markowitz) $\rightarrow$ Binary Quadratic Programming (BQP), NP-hard.
  - Real-world applications: pension funds, long-term portfolios.
- **Quantum Advantage**
  - BQP structure well-suited to quantum hardware.
  - Quantum annealers vs. gate-based quantum computers.
- **Empirical Studies**
  - D-Wave tested on Nikkei225 and S&P500.
  - Hybrid quantum-classical approaches.
  - Benchmarks vs. Gurobi and LocalSolver.

Theoretical Results

Portfolio Optimization: Classical Formulation

- **Inputs:**
  - N assets with returns μ_i, risks σ_i, correlations ρ_{ij}
  - Covariance matrix: $\Sigma = \{\sigma_{ij}\}$ with:
    - $\sigma_{ij} = \sigma_i^2$ if $i = j$, else $\rho_{ij}\sigma_i\sigma_j$
- **Classical Problem:**

$$\begin{aligned} \min \quad & x^T \Sigma x \\ \text{s.t.} \quad & \sum x_i = n, \quad \mu^T x \geq R^*, \quad x_i \in \{0,1\} \end{aligned}$$

  - Select n assets to minimize risk and meet a return target.
- **Towards QUBO:**
  - QUBO = Quadratic Unconstrained Binary Optimization
  - Standard form: $\min x^T Q x$ with $x \in \{0,1\}^n$
  - Constraints encoded via penalty terms

QUBO Reformulation and Ising Mapping

- **Converting Constraints to QUBO:**

$$\min y = c^T x + \lambda (Ax - b)^T (Ax - b) = x^T Q x$$

- **Ising Model (used in quantum annealing):**

$$\min y = \sum h_i s_i + \sum J_{ij} s_i s_j, \quad s_i \in \{-1, 1\}$$

  - Relation: $s_i = 2x_i - 1$ for QUBO-Ising conversion

- **Portfolio QUBO Formulations:**
  - **Equality constraints:**

$$\min \lambda_0 x^T \Sigma x + \lambda_1 \left(\sum x_i - n \right)^2 + \lambda_2 \left(\mu^T x - R^* \right)^2$$

  - **Inequality constraints:**

$$\min \lambda_0 x^T \Sigma x + \lambda_1 \left(\sum x_i - n \right)^2 + \lambda_2 \left(\mu^T x - R^* - \sum 2^k y_k \right)^2$$

Implementation & Results

Determining QUBO Parameters

Finding optimal penalty coefficients:

- Set $\lambda_0 = 1$ (scaling factor)
- λ_1 controls budget constraint penalty
- λ_2 controls return constraint penalty

Rule of thumb:

- Gain from violating constraint must be lower than cost
- Values too low: constraints violated
- Values too high: search efficiency decreased

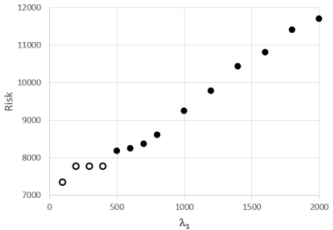

Figure: Relation between Risk and setting of λ. Open dots are violating the original constraints, closed dots are valid solutions.

Determining QUBO Parameters (continued)

- **Calculation for λ_1:**

$$\lambda_1^c = \max_i \sum_{j=1}^{n} \sigma_{i\{j\}}$$

  where $\sigma_{i\{j\}}$ is the j-th smallest covariance value for asset i

- **Calculation for λ_2:**
  1. A_1 = average difference between smallest n sums $S_i = \sum_{j=1}^{n} \sigma_{i\{j\}}$
  2. A_2 = average positive difference in μ_i between these n stocks
  3. $\lambda_2^c = A_1/A_2$

- **Note:** These values are first estimations and starting points for eventual grid search of optimal parameters

Experiment Setup

- **Objective:** Portfolio optimization (risk minimization under budget and return constraints)
- **Datasets:** Nikkei225 and S&P500
- **Compared Methods:**
  - **Quantum:** D-Wave HQPU
  - **Classical:** Simulated Annealing (SA), Genetic Algorithm (GA), Gurobi (GB), LocalSolver (LS)
- **Parameter Grid:** Stock pool size N, selection size n, minimum return R^*

Solution Quality:

- **Small instances ($N \leq 100$):** All methods (except GA) reached optimal or near-optimal solutions
- **Medium ($100 < N \leq 200$):** HQPU/SA within 5% of best, LS best, GA poor, GB memory issues
- **Large ($N > 200$):** LS best, HQPU/SA still competitive, GA poor, GB not applicable

Computational Performance & Observations

- **Computation Time (N=50 to N=500):**
  - **HQPU:** Fastest (1–6s), consistent across instance sizes
  - **SA:** Slower (2–135s)
  - **GA:** Slowest (53–221s)
  - **GB:** Very fast but crashes on large instances
  - **LS:** Efficient but variable (2–207s)
- **Key Insights:**
  - HQPU is suitable for large-scale, time-sensitive tasks
  - LocalSolver gives best quality, but no optimality proof
  - Gurobi best for small instances when proof of optimality is required
  - GA underperforms both in quality and speed

Conclusions & Insights

Strengths of D-Wave HQPU:

- Fast solution time
- Minimal scaling with problem size
- Near-optimal solutions for smaller instances
- Reasonable solutions for larger instances

Limitations & Considerations:

- Solution quality gap for largest instances
- Optimality not proven (unlike Gurobi)
- Parameter tuning required (λ_1 and λ_2)
- Multiple runs needed due to stochastic nature

Best approach by use case:

- **Need for provable optimality:** Gurobi (for $N \leq 150$)
- **Large problems with time constraints:** HQPU
- **Best solution quality:** LocalSolver (though cannot prove optimality in reasonable time)

Replication of the Results

Objective and Methodological Framework

This study replicates and extends Phillipson Bhatia's work on portfolio optimization using quantum computing. The problem is formulated as a QUBO (Quadratic Unconstrained Binary Optimization) and tackled via hybrid quantum-classical solvers.

- **Problem**: Optimize portfolio allocation by minimizing risk under budget and return constraints.
- **Quantum Algorithm**: QAOA (Quantum Approximate Optimization Algorithm), chosen over VQE for its strength in discrete optimization.
- **Classical Optimizer**: COBYLA.
- **Hybrid Approach**: We combine quantum sampling with classical post-processing to mimic D-Wave's capabilities using Qiskit simulators.
- **Exploration**: We varied QAOA parameters such as circuit depth and iterations to study convergence behavior.

Experimental Results and Insights

- **QAOA Performance**: Successfully found optimal or near-optimal solutions on small instances; however, underperformed in return maximization compared to Gurobi and LocalSolver.
- **Comparison with Original Study**: Our simulation provided similar feasible regions, but QAOA results were more sensitive to tuning (e.g., number of layers, learning rate).
- **Hybrid Solver Impact**: Enhanced flexibility allowed tuning across classical and quantum stages, though runtime scaled significantly with problem size.
- **Takeaway**: While classical solvers remain superior for precision, QAOA shows promise in scalability and hybrid strategy potential.

Code Architecture and Simulation Pipeline

Our implementation follows a modular structure to allow reproducibility and experimentation with quantum and classical components.

- **Data Preparation**: Historical asset returns and covariance matrices were used to define the QUBO objective.
- **QUBO Formulation**: Encodes risk-return tradeoff, budget, and return constraints into a binary quadratic model.
- **Quantum Execution**: QAOA circuits constructed with Qiskit, optimized with COBYLA. Parameter sweeps allow exploration of circuit depth and optimizer settings.
- **Comparison Metrics**: Performance assessed by portfolio return, risk (variance), and constraint satisfaction.
- **Visualization**: Results plotted for return-risk space and constraint adherence, allowing direct comparison with classical solvers.

Conclusion

Conclusion: Quantum Portfolio Optimization

- **Quantum computing shows promise** for large-scale portfolio optimization, particularly via hybrid quantum-classical methods.
- **Classical solvers still outperform** quantum approaches in solution quality and constraint handling, especially for return maximization.
- **Key limitations of quantum methods:**
  - Stochastic variability across runs
  - High sensitivity to parameter tuning
  - Difficulty handling return constraints in large problems
- **Future outlook:**
  - Hybrid methods can be useful under time constraints
  - Further development of quantum algorithms like QAOA is needed
  - Classical solvers like Gurobi remain preferable for small, precise tasks

The End