Quantum financial optimization: Application of variational quantum-convex optimization hybrid algorithm to combinatorial optimization problems

November 1, 2024

1 Background

In financial markets, numerous complex mathematical problems are often encountered. This study focuses on the portfolio optimization for stocks. In the field of portfolio optimization, the mean-variance model is commonly used. By utilizing sufficient time-series stock returns data, the mean of the returns is used to predict future stock earnings, and the variance of the returns reflects the volatility of the stock returns. This project adopts the mean-variance model, integrating quantum algorithms with financial contexts, aiming to propose a solution for portfolio optimization problems.

The mean-variance model is formulated as follows, representing a convex optimization problem:

$$\begin{aligned} &\max_{w} \left(w^T r - \gamma w^T V w\right) \\ &\text{s.t. } e^T w = 1, \\ &0 \le w \le 1, \end{aligned} \tag{1}$$

where w is a k-dimensional vector representing the weight, r is a k-dimensional vector representing the expected return, γ is the risk aversion coefficient, and V is a $k*k$ matrix representing the variance.

2 Principle

The advantage of quantum algorithms lies in their powerful parallel computing capabilities. However, they face difficulties in direct dealing with convex optimization problems, which are quadratic mathematical issues. Therefore, an approach is employed where classical algorithms like ADMM are first utilized to transform convex optimization problems into linear problems that are more amenable to quantum algorithms. Then, the variational quantum method is used to solve these linear problems, thereby obtaining solutions to the convex optimization problems.

2.1 Convex Optimization Algorithm: Alternating Direction Method of Multipliers (ADMM)

2.1.1 Original Formula

$$\begin{aligned} &\min_{w,z} \left(f(x) + g(z)\right) \\ &\text{s.t. } AX + Bz = c, \end{aligned} \tag{2}$$

where f and g are convex functions, A and B are matrices, and c is vectors.

The augmented Lagrange function of problem 2 is

$$L_\rho(x, y, z) = f(x) + g(z) + \lambda^T (AX + Bz - c) + \frac{\rho}{2} \cdot \|AX + Bz - c\|_2^2 \tag{3}$$

where λ is a vector representing the scale of the Lagrangian dual variable, and ρ is a positive penalty coefficient defined by the user to control the strength of the penalty term, which is actually the only penalty term.

By defining $\mu = \rho^{-1}\lambda$, ADMM uses the following iterative method:

$$\begin{aligned} x^{k+1} &= \arg\min_{x} f(x) + \frac{\rho}{2} \cdot \|Ax + Bz^k - c + \mu^k\|_2^2 \\ z^{k+1} &= \arg\min_{z} f(z) + \frac{\rho}{2} \cdot \|Ax^{k+1} + Bz - c + \mu^k\|_2^2 \\ \mu^{k+1} &= \mu^k + Ax^{k+1} + Bz^{k+1} - c \end{aligned} \tag{4}$$

where x^k and z^k are the decisive variables of the kth iteration. The meaning of these three iterative steps is: update x, guaranteeing that x obtained makes the sum of $f(x)$ and penalty terms minimum; update z, guaranteeing that z obtained makes the sum of $g(z)$ and penalty items minimum; update μ, the Lagrange multiplier, involving adding the current residual $Ax^{k+1} + Bz^{k+1} - c$ to the current Lagrange multiplier so that μ is adjusted based on the residual of the current iteration.

Define $r^k = Ax^k + Bz^k - c$ as the original residual in the kth iteration, $s^k = \rho A^T B(z^k - z^{k-1})$ as the dual residuals in the kth iteration. The criterion for terminating the iteration is that original and dual residuals are less than the user-defined stopping tolerances $\epsilon^p$and ϵ^d, respectively.

2.1.2 Solving Convex Optimization Problems with ADMM

The objective function of the convex optimization problem focused on in this project includes a quadratic term (variance term) and a linear term (expected return term). It is acceptable to set the expressions of $f(x)$ and $g(z)$ as shown in the formula 5, which exactly meets the requirements of the objective function and constraints of the convex optimization problem

$$\begin{gathered} f(x) = \frac{1}{2} x^T Q x + x^T p \\ s.t. Ax = b \end{gathered} \tag{5}$$

$$g(z) = I_{l \leq z \leq u}(z) \tag{6}$$

where $g(z)$ is an indicator function representing a linear inequality constraint. According to $ADMM$, the original problem can be written as:

$$\begin{gathered} min\ (\frac{1}{2} x^T Q x + x^T p + I_{l \leq z \leq u}(z)) \\ s.t.\ x - z = 0 \end{gathered} \tag{7}$$

Applying the iterative method proposed by formula 4 obtains the following formulas:

$$x^{k+1} = \arg\min_{x|Ax=b} \frac{1}{2} x^T Q x + x^T p + \frac{\rho}{2} \cdot \|x - z^k + \mu^k\|_2^2 \tag{8a}$$

$$z^{k+1} = \arg\min_{l \leq z \leq u} \frac{\rho}{2} \cdot \|x^{k+1} - z + \mu^k\|_2^2 \tag{8b}$$

$$\mu^{k+1} = \mu^k + x^{k+1} - z^{k+1} \tag{8c}$$

It is noted that formula 8b is a least squares problem with box constraints, so it can be solved analytically:

$$\Pi(x) = \begin{cases} l, & if\ x < l \\ x, & if\ l \leq x \leq u \\ u, & if\ x > u \end{cases} \tag{9}$$

Therefore, the analytical solution of formula 8b can be written as follows:

$$z^{k+1} = \Pi(x^{k+1} + \mu^k) \tag{10}$$

Formula 8a is a quadratic programming problem with equality constraints, which can also be solved analytically. Specifically, x is the solution to formula 8a when x and λ satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions. The KKT conditions should include two parts: (1) the gradient of Equation 8a with respect to the variable x is zero, and (2) Equation 8a satisfies the constraint conditions on x. Since,

$$L(x^{k+1}, \lambda) = \frac{1}{2}x^T Qx + x^T p + \frac{\rho}{2} \cdot \|x - z^k + \mu^k\|_2^2 - \lambda(Ax - b)$$

Thus, the KKT conditions can be expressed as follows:

$$L_x^{'}(x^{k+1}, \lambda) = Qx + p + \rho(x - z^k + \mu^k) - \lambda A = 0$$
$$Ax = b$$

Simplify a matrix form as follows with $\eta^{k+1} = -\lambda$:

$$\begin{pmatrix} Q + \rho I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} \\ \eta^{k+1} \end{pmatrix} = - \begin{pmatrix} p - \rho(z^k - \mu^k) \\ -b \end{pmatrix} \tag{11}$$

That is, a function of the form 5, 6 can be solved by the following iterative formula:

$$\begin{pmatrix} x^{k+1} \\ \eta^{k+1} \end{pmatrix} = - \begin{pmatrix} Q + \rho I & A^T \\ A & 0 \end{pmatrix}^{-1} \begin{pmatrix} p - \rho(z^k - \mu^k) \\ -b \end{pmatrix} \tag{12a}$$
$$z^{k+1} = \Pi(x^{k+1} + \mu^k) \tag{12b}$$
$$\mu^{k+1} = \mu^k + x^{k+1} - z^{k+1} \tag{12c}$$

If ρ is static, the coefficient matrix in 12a remains the same in each iteration and is decomposed only once at the beginning of the ADMM algorithm. Such a feature is very helpful for improving computing efficiency.

Now compare this convex optimization problem with the mean-variance model:

$$min\ f(x) = \frac{1}{2}x^T Qx + x^T p$$
$$g(z) = I_{l \le z \le u}(z)$$
$$s.t. Ax = b$$

$$\min_w \gamma w^T V w - w^T r \tag{13}$$
$$\text{s.t. } e^T w = 1,$$
$$0 \le w \le 1,$$

To apply the conclusion of the formula 12, the following transformation should be made:

$$Q = 2\gamma V \tag{14a}$$
$$p = -r \tag{14b}$$
$$A = e^T \tag{14c}$$
$$b = 1 \tag{14d}$$
$$l = 0 \tag{14e}$$
$$u = 1 \tag{14f}$$

Bring to the results as follows:

$$\begin{pmatrix} x^{k+1} \\ \eta^{k+1} \end{pmatrix} = -\begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} -r - \rho(z^k - \mu^k) \\ -1 \end{pmatrix} = \begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} r + \rho(z^k - \mu^k) \\ 1 \end{pmatrix} \tag{15a}$$

$$z^{k+1} = \Pi(x^{k+1} + \mu^k) \tag{15b}$$

$$\mu^{k+1} = \mu^k + x^{k+1} - z^{k+1} \tag{15c}$$

Define $M = \begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}$ $b = \begin{pmatrix} r + \rho(z^k - \mu^k) \\ 1 \end{pmatrix}$

Since M is a constant and real symmetric matrix, it suffices to compute M^{-1} only once before the iteration starts, allowing us to focus on the iteration among variables. According to the requirements of formula 15, solving for x is necessary to compute the other two variables. Therefore, the current challenge lies in solving the linear system represented by formula 15a.

Next, we attemp to solve this linear system using variational quantum algorithms.

2.2 Variational Quantum Algorithm

Currently, numerous studies have explored quantum algorithms for solving linear problems, with the well-known HHL algorithm specifically designed for this purpose. Taking the linear problem $Mx = b$ as an example, the HHL algorithm addresses a scenario similar to this project, where both the matrix M and the vector b are known, and the unknown vector x is to be solved. However, upon further investigation, it is believed that the HHL algorithm is not applicable to this project for the following reasons:

1. The Hamiltonian simulation used in the HHL algorithm requires the coefficient matrix of the linear system to be sparse, with its sparsity directly related to the algorithm's performance. However, the coefficient matrix in this project is extremely dense, with only one zero value, greatly weakening the potential performance advantage of the quantum algorithm.

2. The HHL algorithm necessitates to know the eigenvectors or the singular matrix of M, which are used when preparing the quantum state $|b\rangle$ as a projection onto the eigenvectors. Given the scale of the project, if the eigenvectors of M are known, it would be more reasonable and convenient to directly solve for the vector x after performing spectral decomposition on M, making the use of the HHL algorithm unnecessary.

In summary, the project adopts a variational quantum algorithm, designing a Hamiltonian operator such that the quantum state to be solved serves as its eigenstate. This transforms the problem of solving the linear system of equations into the problem of finding the eigenstates of the Hamiltonian operator.

2.2.1 Designing the Hamiltonian Operator H

Thinking of the original system of linear equations, if vectors x and b in $Mx = b$ are normalized and constructed as quantum states, the solution to the original problem can be expressed as $|x\rangle = \frac{M^{-1}|b\rangle}{||M^{-1}|b\rangle||}$. Designing the Hamiltonian operator $H = M^\dagger(i - |b\rangle\langle b|)M$, $|x\rangle$ is an eigenstate of H with an eigenvalue of zero, since:

$$\begin{aligned} \langle x|H|x\rangle &= \frac{\langle b|M^{-\dagger}}{||\langle b|M^{-\dagger}||} M^\dagger(I - |b\rangle\langle b|)M \frac{M^{-1}|b\rangle}{||M^{-1}|b\rangle||} \\ &= \frac{\langle b|(I - |b\rangle\langle b|)|b\rangle}{||M^{-1}|b\rangle||^2} \\ &= \frac{\langle b|b\rangle - \langle b|b\rangle\langle b|b\rangle}{||M^{-1}|b\rangle||^2} \\ &= \frac{\langle b|b\rangle - \langle b|b\rangle}{||M^{-1}|b\rangle||^2} \\ &= 0 \end{aligned} \tag{16}$$

Also, M is a Hermitian matrix, So $\langle x|H|x\rangle = \langle x|M(I-|b\rangle\langle b|)M|x\rangle = \sum\limits_{i}|\langle x|M|x_i\rangle|^2 \geq 0$, where, $I - |\ rangle\ |langleb = b\ undersetI\ sum|x_I\ rangle\ langlex_I|$, $langlex_I|x_j\ rangle =\ delta_{I,j}$. So the Hamiltonian operator H is a semi-positive definite matrix, where $|x\rangle$ is also the ground state of H. The task of solving a system of linear equations is transformed into the task of finding the ground state of the Hamiltonian operator.

2.2.2 Designing the Quantum Circuit Diagram

In order to find the ground state of Hamiltonian by variational quantum algorithm, we should first simulate a quantum state $|x\rangle = \frac{M^{-1}|b\rangle}{||M^{-1}|b\rangle||}$ by using parametric quantum circuit, express the Hamiltonian operator and calculate its expectation on the simulated quantum state. Since the theoretical expectation of Hamiltonian operator is 0, the loss function of the operator solving the gradient is defined as $L(\theta) = \langle\phi(\theta)|H|\phi(\theta)\rangle$, which is merged into the quantum neural network. The gradient descent method is used to continuously optimize the parameter θ. According to the final optimal parameter θ, the quantum state, which is also the ground state of H, is calculated.

In addition to parametric quantum gates, CNOT gates need to be placed to simulate quantum entanglement between different quantum circuits.

2.3 Summary

In summary, the approach of this project involves first transforming the objective function and constraints into an iterative problem using the classical convex optimization algorithm ADMM. The primary computational task within one iteration is to solve a set of linear equations. To address this, a variational quantum circuit is constructed. Once the solution to the linear equations is obtained, the iterative equations are updated, generating a new vector b, and a new set of solutions is recalculated. This process continues until the primal and dual residuals of the iterative equations are both less than the stopping tolerance, at which point the iteration terminates, and the quantum state in the quantum circuit represents the final result.

3 Results

This project obtained the return data of 15 stocks listed on the NASDAQ stock exchange from Yahoo Finance, spanning from June 1, 2023, to June 30, 2024. These 15 stocks are AKR, AMD, AMKR, AMZN, ANET, AROC, BAC, BAM, BK, BKKT, BKNG, BKR, BLK, CBOE, CHKP, all of which have asset values in the billions of US dollars.

The calculation of expected return and covariance of stock returns refers to Barra model, incorporating the concept of half-life to ensure that more recent return rate data carries greater weight in the mean calculation and has a more significant impact on parameter estimation. Since solving the expected return is not the focus of this project, the return data of June 30, 2024, is used as the expected return of the stocks, and the data before that is used for the calculation of the covariance matrix.

In the quantum part, the covariance matrix V obtained by 15 stocks has the shape of $15*15$. After operated by ADMM algorithm, the coefficient matrix $M = \begin{pmatrix} 2\gamma V + \rho I & e \\ e^T & 0 \end{pmatrix}$ involved in the iteration is a real symmetric matrix with the shape of $16*16$. So the number of quantum circuits required for this project is $n = log_2 16 = 4$.

The quantum part uses a python library, mindquantum, and code runs on Huawei Cloud.

Pparameters used in this project and their descriptions are listed as follows:

3.1 Determination of γ

According to the formula 13, $\gamma$is an important parameter to measure the relative proportion of expected return and risk in the objective function. Therefore, the value of γ should ensured that: (1) the expected return and risk are of the same order of magnitude to prevent the objective function value from being excessively influenced by either term; (2) the objective function value should not be too close to zero, as this may result in insignificant changes due to insufficient computational precision.

Table 1: Parameter and description

Parameter	Description	Value
k	Number of stocks	15
h	Time window for covariance matrix calculation	252
τ	Half-life period	252
n	Number of qubits in the quantum circuit	4
γ	Risk aversion coefficient	15
ρ	Penalty coefficient in the augmented Lagrangian function	1
λ	Lagrange dual variable	
μ	$\rho^{-1}\lambda$	
ϵ^p	Stopping tolerance for primal residual	0.001
ϵ^d	Stopping tolerance for dual residual	0.001

When the weight values of all stocks are equal, the risk aversion coefficient that makes the expected return and risk equal is 10. Taking this into consideration, the final choice is

$$\gamma = 15$$

3.2 Quantum Circuit Diagram

Since the M is the size of $16 * 16$, a shallow quantum circuit is constructed in the quantum state of $n = 4$ qubits to achieve the approximate encoding of the quantum state of $|x\rangle$. There are 16 $RY(\theta)$ gates and 7 CNOT gates. It should be noted that the four $RX(\theta)$ gates in the circuit are encoder parts, whose parameters are 0 from beginning to end and do not participate in variational quantum calculation. So they are equivalent to four $RX(0)$ gates, which do not produce any changes to the quantum state.

The initial parametric quantum gate circuit is shown in Figure 1.

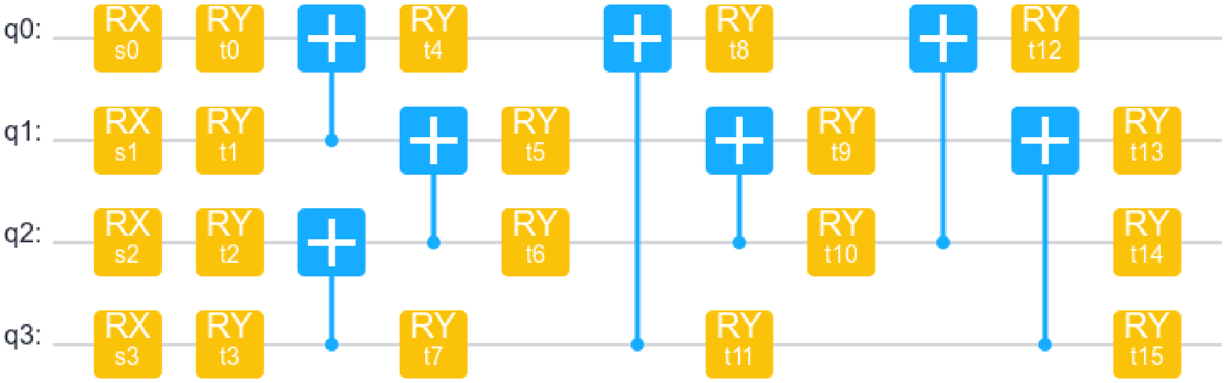

Figure 1: Quantum circuit diagram with parameters

3.3 Number of Iterations

Define the stop tolerance of original residuals as $\epsilon^p = 0.001$ and stop tolerance of dual residuals as $\epsilon^d = 0.001$, ensuring that iteration is continued until both the original and dual residuals are less than the stop tolerance. There are 121 iterations in total, so there are 121 points on both the graph 2 and the graph 3.

3.3.1 Expectation Value of the Hamiltonian Operator

The Hamiltonian operator acts on the circuit diagram shown in the graph 1 in the form of a sparse matrix, yielding an expected value in each iteration. According to the principle in chapter 2.2.1, this theoretical expectation is 0. Figure 2 depicts the expected values of the Hamiltonian operator in successive iterations. Only two iterations deviate from the theoretical value by more than 0.07%. This demonstrates the correctness of the theory employed in this project and the validity of the obtained results.

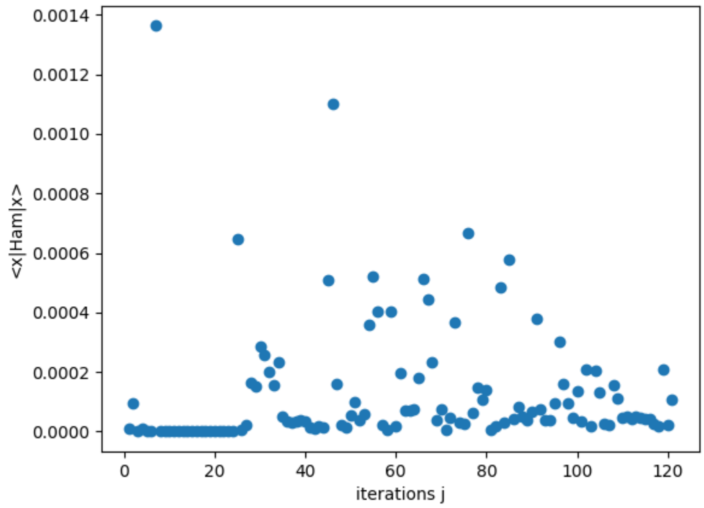

Figure 2: Expectation value of Hamiltonian operator

3.3.2 Objective Function Value $\gamma w^T V w - w^T r$

The solutions to the linear equations calculated in each iteration is substituted into the objective function, and the change in the objective function is shown in figure 3. The function exhibited an overall downward trend until it eventually stabilized, which aligns with the trend of the ADMM algorithm in searching for the minimum value of the objective function as described in chapter 2.1.2.

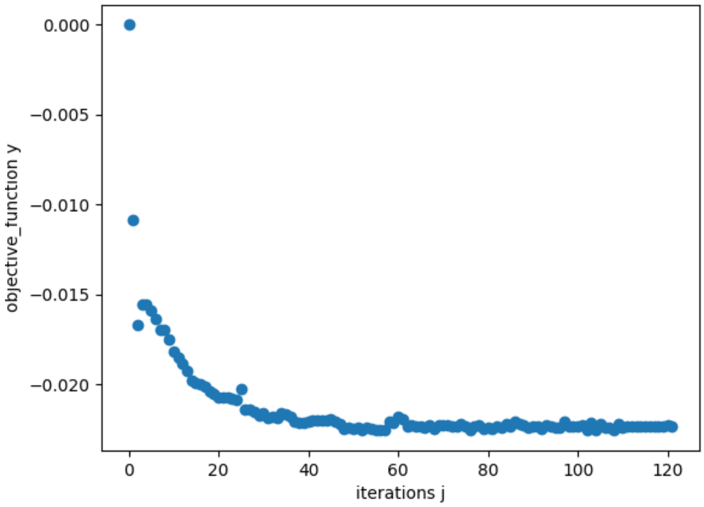

Figure 3: Objective function value

3.4 Compare with the result of classical algorithm

After all calculations are completed, the optimal parameters of each quantum gate in the obtained quantum circuit diagram are shown in the figure 4.

The optimal weights of stocks obtained according to these parameters are shown in the second column of table 2. Some of the weights are negative. Considering that the weight value of $[0, 1]$ has

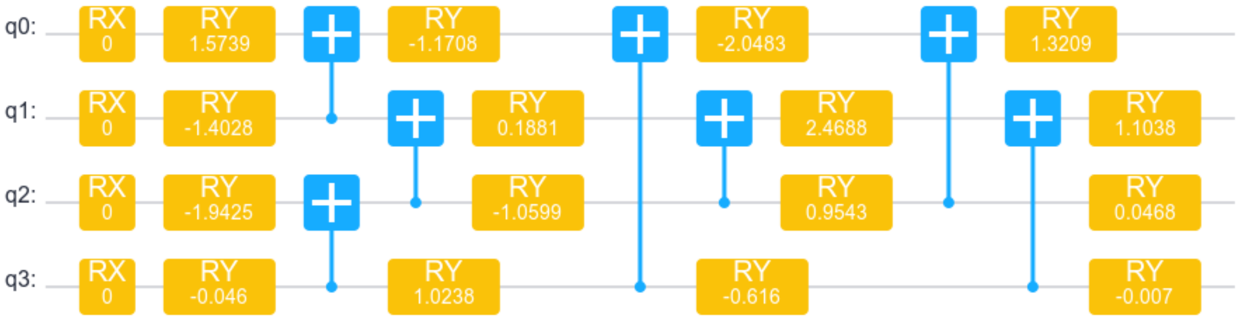

Figure 4: Quantum circuit diagram with optimal parameters

been incorporated in the construction of the iteration, it is speculated that the accumulated errors in the calculation process will eventually show negative weight and the quantum related programs are incapable of distinguish them. By setting the negative weights to zero and then re-normalizing, the resulting weights are presented in the third column of table 2. The fourth column shows the weight obtained using the *cvxpy* library, which employs a classical convex optimization algorithm. Since the second column represents the actual results from the quantum-classical algorithm, the analysis is based on these values.

Comparing the objective function values of the two methods, the quantum-classical algorithm yields a slightly higher value, with a deviation of 0.16% from the classical algorithm. Comparing the weight data of the two methods, it is found that both algorithms concentrate their selections on stocks 0, 2 and 11 (AKR, AMKR, and BKR, respectively). This indicates that the initial selection of 15 stocks and the parameter settings (such as γ) in this project have significant limitations. The expected returns and stability of these three stocks are apparently better than the others, and the contributions of the other stocks to the final result of the objective function are relatively low. Consequently, the portfolio optimization is almost exclusively conducted among these three stocks, failing to demonstrate the superiority of diversification.

Table 2: Comparison of calculation results between quantum-classical algorithm and classical method

Serial No.	Weights from Quantum-Classical Algorithm	Non-negative Weights from Quantum-Classical Algorithm	Weights from Classical Algorithm
0	2.43929843e-01	2.42882733e-01	1.92118776e-01
1	-1.48296628e-03	0	2.10135440e-22
2	4.80616983e-01	4.78553853e-01	5.25081379e-01
3	-6.96886244e-04	0	7.49192735e-23
4	5.56502021e-05	5.54113142e-05	9.69256530e-23
5	2.69331050e-03	2.68174901e-03	1.48358147e-22
6	-4.21995417e-04	0	9.86542067e-23
7	3.87218790e-03	3.85556588e-03	6.50809847e-23
8	-2.43143670e-04	0	1.67722455e-22
9	-3.99785463e-04	0	-6.30689551e-23
10	-9.07155027e-04	0	7.13622351e-23
11	2.72523170e-01	2.71353318e-01	2.82799845e-01
12	6.20030959e-04	6.17369371e-04	1.64311697e-22
13	-1.51856999e-04	0	6.53047173e-23
14	-7.38576272e-06	0	6.70058343e-23
$\gamma w^T V w - w^T r$	-0.02232539049260529	-0.022278119863211213	-0.022361270760996213

4 Discussion

The project combines the Alternating Direction Method of Multipliers (ADMM) with variational quantum algorithms to solve the portfolio optimization problem in the stock market. First, ADMM transforms the portfolio optimization problem into an iterative solution for a system of linear equations Then a variational quantum algorithm constructs a neural network to solve this system. The quantum state in the quantum circuit at the end of the iteration represents the solution to the portfolio optimization problem. Comparing the final weights obtained with those calculated using classical methods, it is noted that the objective function values and weights are generally consistent, but the computational accuracy of the hybrid quantum-classical algorithm is far inferior to the classical method, resulting in many negative values. In practical applications, these negative values should be set to zero and the weights should be re-normalized.

Upon summarization, I think the following issues in this project deserve further exploration:

1. Representation of the dense coefficient matrix M. At present, the relatively mature quantum method for solving linear systems of equations, known as HHL algorithm, requires the coefficient matrix to be sparse and the degree of quantum acceleration is closely related to the sparsity. This is mainly because the Hamiltonian simulation used in HHL. In this project, the coefficient matrix M is exceptionally dense, so HHL is not available. I did not think of a reasonable solution, instead, I chose to transform it into a sparse matrix representation and directly use in the calculation. The representation of sparse matrix is allowed to be directly used in the construction of Hamiltonian operators in mindquantum library. In addition, the quantum scale of this project is small. Therefore, I have successfully realized the calculation of dense sparse matrix in this project, but a more general algorithm about it needs to be further explored.

2. More suitable integration of ADMM (or other convex optimization methods) with Quantum Algorithms. A major advantage of using ADMM iterative method to solve portfolio optimization problems is that the coefficient matrix is constant and only the vector on the right side needs to be recalculated in each iteration. However, the Hamiltonian operator in this project has M and b at the same time. In practise, I still recalculated the Hamiltonian matrix in each iteration, failing to leverage the advantages of the ADMM algorithm.