UNIVERSIDAD EAFIT

Vigilada Mineducación

# OPTIMIZACIÓN DE CARTERA DE ACTIVOS FINANCIEROS UTILIZANDO MARKOWITZ Y BLACK-LITTERMAN: UNA PERSPECTIVA DESDE LA COMPUTACIÓN CUÁNTICA

Portfolio Optimization of Financial Assets Using Markowitz and Black-Litterman: A Perspective from Quantum Computing

CARLOS ANDRÉS JARAMILLO PINEDA

Trabajo de grado para optar al título de Magíster en Ciencia de Datos y Analítica

Asesores:

Paula María Almonacid Hurtado
palmona1@eafit.edu.co

Juan Guillermo Lalinde Pulido
jlalinde@eafit.edu.co

UNIVERSIDAD EAFIT

ESCUELA DE CIENCIAS

MAESTRÍA EN CIENCIAS DE LOS DATOS Y LA ANALÍTICA

MEDELLÍN

2024

**Carlos Andres Jaramillo Pineda**

EAFIT University, School of Applied Sciences and Engineering-master's in data science and Analytics (Antioquia)-Medellin, Colombia

* Correspondence: cajaramilp@eafit.edu.co

**Abstract:**

Quantum computing, currently in its emerging stage, holds the potential to revolutionize various sectors, including finance. While portfolio optimization strategies based on classical methods like Markowitz and Black-Litterman have already proven effective, the introduction of quantum algorithms could significantly enhance these techniques in terms of predictive ability and computational efficiency. Building upon previous research such as that by Bova (2021), which underlines the crucial role that quantum computing could play given the volume and complexity of financial data, this paper proposes a framework that integrates classical Markowitz and Black-Litterman theories with quantum computing. Through this hybrid approach, we explore how hybrid classic-quantum algorithms approaches can enrich the portfolio optimization process, offering significant advantages in financial analysis and strategic decision-making.

**Keywords:**

Hybrid quantum-classic algorithms, portfolio optimization, Markowitz, Black-Litterman, quantum finance.

**Objectives**

**General Objective:**

The primary objective of this academic research is to conduct an evaluation of classical portfolio optimization techniques, specifically the Markowitz and Black-Litterman models, under various stock market scenarios, and to complement these techniques with more recent approaches by applying a hybrid model that integrates classical and quantum algorithms, such as the Quantum Approximate Optimization Algorithm (QAOA).

**Specific Objectives:**

To conduct a review of the literature relating to classical portfolio optimization methods, with a focus on the Markowitz and Black-Litterman models, as they apply to stock portfolio selection.

To apply the Quantum Approximate Optimization Algorithm (QAOA), a hybrid quantum-classical approach, to portfolio optimization, and compare its performance against the classical methods.

To simulate the previous methods in different market scenarios with varying the market conditions to test the robustness of the classical optimization methods.

To implement the QAOA method and evaluate its efficiency and effectiveness in optimizing stock portfolios compared to classical optimization techniques.

To analyze the results to determine the practical advantages or limitations of employing a hybrid quantum approach like QAOA in real-world financial scenarios.

To explore the potential for integration of hybrid quantum-classical algorithms in the field of financial decision-making.

## 1. Introduction

Optimization problems play a crucial role in various areas of science and engineering, and the financial sector is not an exception. The optimization of portfolios of financial assets presents itself as a computationally intense challenge, especially in scenarios where multiple variables and constraints are considered. These optimization problems, when all constrains are taking into account, are NP-hard problems, a category of problems for which no solution is known that can be obtained in reasonable computing times. In portfolio optimization having N assets results in a total of $2^N$ possibilities of portfolios. When dealing with a list of assets or stocks as limited as forty, this becomes impractical to handle on a regular computer. (Cohen, 2020)

The classic paradigm in portfolio optimization was established by Harry Markowitz in the 1950s, under the hypothesis that investors are risk-averse and, therefore, seek portfolios that offer the maximum expected return for a given level of risk. Later, the Black-Litterman model was introduced as an extension that integrates the investor's subjective opinions and updates the assets' expected returns. However, although these models have proven their utility, they are limited by current computational capabilities when dealing with large asset sets and market scenarios.

With the arrival of quantum computing, the possibility opens to approach these optimization problems from a new perspective. Quantum algorithms, and hybrid versions such as the Quantum Approximate Optimization Algorithm (QAOA), show great potential for solving NP-hard problems more efficiently than their classical counterparts, or at least having better approximations. This quantum-classical hybrid algorithm may offer significant advantages in terms of calculation time and solution accuracy.

The overall purpose of this study is to explore the optimization of portfolios of financial assets using the classic methods of Markowitz and Black-Litterman, and evaluate the advantages and challenges associated with the implementation of the QAOA algorithm in this context.

While various types of algorithms exist for addressing intractable problems (Cohen, 2020), classical computers often struggle to execute them efficiently when dealing with realistic inputs. The objective of this study is to utilize well-known methods such as the Markowitz and Black-Litterman approaches to allocate weights in a portfolio and investigate whether methods like QAOA can serve as valuable tools to assist investors in making data-driven decisions when selecting assets for portfolio optimization.

We seek to understand whether quantum computers, and hybrid approaches executed on classical computers, despite being in an initial stage of development, can offer practical and efficient solutions for portfolio optimization that are comparable to or superior to conventional techniques. The QAOA approximation has the advantage of running simulations on both classical and quantum computers.

In summary, this article positions itself at the intersection of quantitative finance and quantum computing, seeking to open new horizons in financial optimization using emerging quantum technologies.

## 2. Methods

In many fields of knowledge, solving problems often involves identifying the points at which a function reaches its maximum or minimum values. While, as González (2021) highlights, there are various numerical calculation tools available to approximate the maxima and minima of a function under certain constraints, such as continuity and differentiability of the function, but many practical applications across different industries exclude the use of these numerical methods due to the function's complexity. Problems of this nature are recognized as combinatorial optimization, where locating the minima and maxima of a function is deemed an NP-hard problem, making finding the optimal solution computationally infeasible for real data.

In the vast landscape of quantitative finance, the optimization of investment portfolios represents a challenging task. Since the introduction of modern portfolio theory by Harry Markowitz in 1952, and its subsequent refinement with models like that of Black and Litterman in 1992, the financial community has employed various methods to improve the efficiency of their investments. These methods, although powerful, are not exempt from limitations—primarily computational—that diminish their efficacy in increasingly complex market scenarios (Gunjan, 2022).

In this context, quantum computing emerges as a promising technology with the potential to address NP-hard computational problems, including those in finance. With the arrival of the quantum paradigm, enhancing classical portfolio optimization methods becomes a promising prospect.

The overall aim of this article is to evaluate the effectiveness of financial portfolio optimization methods using the Markowitz and Black-Litterman models as comparative bases. For this, various financial assets that are part of the S&P 500 index will be selected and initially, the classical methods mentioned will be applied to determine their efficiency in terms of profitability and risk. Subsequently, the QAOA algorithm (Quantum Approximate Optimization Algorithm) implemented in Qiskit will be introduced; this is a hybrid quantum-classical variational algorithm designed to address combinatorial optimization problems, with the purpose of comparing its performance with the approaches of Markowitz and Black-Litterman.

According to research conducted by Montoya (2016), classic approaches to portfolio optimization have a rich history, dating back to the publication of the first paper by Markowitz (1952), thereby establishing it as one of the most widely utilized models. Additionally, since 1992, the Black-Litterman optimization method has been available, which incorporates investors' perspectives to predict asset behavior based on expert criteria.

In terms of portfolio optimization, the primary objectives are to minimize risk and maximize expected returns, which can be measured using the portfolio's standard deviation and the expected returns of the assets. To achieve this, the correlation between different assets is used, expressed through a matrix of variances and covariances, along with the expected returns.

Markowitz (1952) emphasized the importance of diversification in portfolio composition, and his model has been improved upon by other authors, such as Sharpe (1963), who developed the "The Diagonal Model." Even though the model's outcomes are fragmented, they efficiently represent the relationship between assets (Montoya, 2016).

However, on the other hand, Markowitz's model has been criticized for the issue of corner solutions and the estimation of returns, which rely on the past performance of assets.

To improve upon Markowitz's method, Robert Litterman and Fischer Black developed an asset distribution model for portfolio management in September 1992, known as the Black-Litterman Global Asset Allocation Model, which was published in the Financial Analyst Journal (Black & Litterman, 1992). This methodology is an extension of Markowitz's model, in which investors with initial capital maximize expected utility while controlling risk. (Montoya, 2016).

In a study conducted by Franco and Avendaño (2011), they described the Black-Litterman Model (BLM), which is based on a market equilibrium situation where return expectations are balanced with the supply and demand for financial assets, assuming that, all investors have the same expectations. In the BLM, if an investor's expectations do not differ from the market's, it is not necessary to specify a return for each asset, as the equilibrium returns corresponding to each asset in the model are used. The next step is to obtain expected returns through inverse optimization; that is, instead of asking what weight is needed to achieve a certain return, it considers what expected return is obtained with the weight indicated by capitalization. Once the expected return is calculated, the model incorporates one of its most significant contributions, which is the inclusion of the investor's market expectations. An expectation is an assumption about the future, which may or may not be realistic. In the case of an investment portfolio, it refers to prospects or expectations about the future performance of a particular security or sector. Additionally, a level of confidence is specified, which represents the prior probability that this expectation will be met according to the investor.

**2.1 Classic Approach: Markowitz Model**

According to Besslera & Opfer (2017), in the Markowitz method (1952), the investor aims to optimize a balance between the risk and return of the investment portfolio. The optimization problem centers on mean-variance, with the equation shown as (1).

$$\max_{w} U = \omega' \mu - \frac{\delta}{2} w' \sum w \quad (1)$$

Where U is the investor's utility, μ is the vector of expected return estimates and δ is the risk aversion coefficient. Additionally, a budget constraint must be included to ensure that the sum of the asset weights or allocations equals 1, expressed as equation (2).

$$\sum_{i=1}^{N} w_i = 1 \quad (2)$$

Where $\omega_i$ is the weight of asset i in the portfolio, and N is the number of assets in the portfolio.

Moreover, equation (3) specifies that for i=1,…,N.

$$w_i \leq 1 \quad (3)$$

Additionally, the portfolio's maximum volatility must be limited to distinguish between different types of investors in terms of their desired maximum level of portfolio risk, shown as equation (4).

$$\sqrt{w'\Sigma w} \leq \hat{\sigma} p \quad (4)$$

Where $\omega$ is the portfolio weight vector, Σ is the covariance matrix of asset returns, and $\hat{\sigma} p$ is a pre-defined portfolio volatility constraint. The volatility constraint $\hat{\sigma} p$ represents an upper limit on volatility rather than a target volatility for the optimized portfolio.

Markowitz's optimization framework assumes that returns are normally distributed, or that mean-variance preferences exist, focusing solely on the mean and variance of returns and ignoring higher moments. Although this seems like a critical assumption, Landsman and Nešlehová (2008) demonstrate that it is sufficient for returns to be elliptically symmetrically distributed for all investor preferences to be equivalent to mean-variance preferences. In the Markowitz method, the sample mean $\mu$ and the sample covariance matrix Σ are used, as per equation (1). Additionally, a budget constraint is necessary, such as prohibiting short sales and limiting the maximum allowed portfolio volatility.

**2.1.1 Characteristics of the Efficient Frontier**

The efficient frontier is a central concept in modern portfolio theory, proposed by Harry Markowitz in 1952. The basic idea is that, given a set of financial assets with different risk and return profiles, there exists a set of portfolios that offer the maximum expected return for a given level of risk or, alternatively, the minimum risk for a given level of expected return.

- Return and risk: The vertical axis typically represents the portfolio's expected return, while the horizontal axis represents the portfolio's risk, commonly measured as the standard deviation of returns.
- Best options: Portfolios on the efficient frontier are theoretically the best options available.

- Diversification: Portfolios on the efficient frontier are diversified in such a way that unsystematic or asset-specific risk is minimized.

### 2.1.2 Limitations of the Mean-Variance Portfolio

- Assumes that stocks are always liquid. That is, stocks can be bought and sold at any time.
- It does not account for transaction costs and administrative fees.
- Statistical limitations. It assumes that returns are independent and identically distributed and can only be expressed through the covariance matrix.
- Dividend payments are not considered when calculating returns.

## 2.2 Classic Approach: The Black-Litterman Model

The Black-Litterman optimization method allows for the inclusion of estimation errors in return forecasts. According to Besslera & Opfer (2017), it is possible to combine two sources of information: "neutral" returns and "subjective" return estimates, also known as "views." One of the advantages of the Black-Litterman approach is that investors can provide return estimates for each asset or maintain a neutral stance for those assets for which they do not feel comfortable making return forecasts.

The implied returns used in the original Black-Litterman approach (Black & Litterman, 1992) are derived through reverse optimization, assuming the observable market or reference portfolio weights $\omega^*$ are the result of a mean-variance optimization (Besslera & Opfer , 2017).

Implied returns are calculated as follows:

$$\Pi = \delta \sum w^* \qquad (5)$$

Where Π is the vector of implied excess returns for the assets, Σ is the covariance matrix, $\delta$ is the investor's risk aversion coefficient, and $\omega^*$ is the vector of observable market or reference portfolio weights.

The Black-Litterman model combines the vector of implied returns Π with the investor's "views" expressed in vector Q. The reliability of each "view" is quantified in a matrix Ω. The combined return estimates are expressed as equation 6:

$$\widehat{\mu_{BL}} = [(\tau\sum)^{-1} + P'\Omega^{-1}P)]^{-1}[\tau\sum)^{-1}]\Pi P'\Omega^{-1}Q \qquad (6)$$

Where $P$ is a binary matrix containing information on which asset has a subjective return estimate and $\tau$ is a factor measuring the reliability of the implied return estimates.

The combined return estimate is a matrix-weighted average of the implied returns and the "views," considering the correlation structure (Lee, 2000). The weighting factors are the measures of uncertainty of the implied returns $\tau$ and the subjective return estimates Ω (Besslera & Opfer , 2017).

The posterior covariance matrix (Satchell and Scowcroft 2000) is calculated as equation 7:

$$\Sigma_{BL} = \Sigma + [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1} \qquad (7)$$

After calculating the combined return estimates and the posterior covariance matrix, a traditional risk-return optimization is performed, maximizing the investor's utility function as presented in Equation 1 of the Markowitz model.

$$\max_{w} = w'\mu - \frac{\delta}{2} w'\Sigma w \qquad (8)$$

In conclusion, the Black-Litterman model, complements the Markowitz approach by incorporating investor views and handling estimation errors in return forecasts.

### 2.2.1 Limitations of the Black-Litterman Portfolio

1. **Subjectivity in views**: As pointed out by Stoilov (2020), the model can be sensitive to the subjective views or opinions used in the model. This creates a level of inconsistency in portfolio recommendations when different experts provide their views, as these don't have a standardized base for comparison.
2. **Complexity and computational intensity**: The Black-Litterman model involves a more complex mathematical framework compared to classical models, which could be computationally intensive.
3. **Reliance on equilibrium market weights**: The model is dependent on market equilibrium weights, and if the market is not efficient or if there are anomalies, the results might be skewed.
4. **Estimation risk**: Like the Markowitz model, Black-Litterman is also susceptible to estimation risks related to inputs like expected returns and the covariance matrix.

## 2.3 Quantum Computing

There are optimization problems that are very difficult to solve in a reasonable amount of time using traditional computing. However, quantum computing has the potential to overcome these barriers. Although quantum computers are still in development and are not yet free from noise that limits their application, cloud-based solutions like those from IBM have already demonstrated significant capabilities in solving optimization problems, according to Sahin (2021).

To tackle the proposed problem of portfolio optimization with a hybrid approach between classic and quantum computing, it is necessary to outline the key terms in the context of quantum mechanics and quantum computing.

As Sotelo (2021) notes, unlike the bits in classical computing, quantum computers use qubits as the minimal unit of information. A qubit has two base states, which can be mathematically represented by two vectors and combined linearly. In a state of superposition, qubits can represent multiple values simultaneously, owing to the principle of superposition in quantum theory.

The qubit has two base states, which can be noted as two vectors as in:

$$\text{State } 0 : \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \text{and} \qquad \text{State } 1 : \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

For example, in a classical bit, the value can be either 0 or 1. But in a qubit in a state of superposition, it can be a superposition of 0 and 1 at the same time. This is because qubits operate according to the laws of quantum mechanics and can exist in a superposition of multiple states simultaneously.

Superposition allows quantum computers to perform calculations in parallel, which is useful for solving complex problems more efficiently than classical computers. In portfolio optimization, for instance, qubits in a state of superposition can represent different combinations of assets and their weights in the portfolio simultaneously. This allows for quicker evaluation of multiple possible solutions, according to Kuchkovsky (2022).

Furthermore, the qubit can also be a linear combination (with complex coefficients) as follows:

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \alpha_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

where $\alpha_0$ and $\alpha_1$ are complex numbers and $|\alpha_0|^2 + |\alpha_1|^2 = 1$

**2.3.1 Gates**

Another key concept in quantum computing are gates. According to Sotelo (2021), in quantum computing, a gate is an operator that acts on one or more qubits to change their state. This mirrors classical computing where logical gates are used to manipulate bits, in quantum computing, gates facilitate logical and arithmetic operations on qubits.

Basic quantum gates include the Hadamard, the SWAP and the CNOT gate. The Hadamard gate, for instance, is used to create superpositions and is represented by the letter H.

Each quantum gate has a specific mathematical representation in terms of unitary matrices and can be combined with other gates to perform more complex operations on qubits. The ability to build quantum circuits from gates and combine them flexibly is one of the main advantages of quantum computing. (Sotelo, 2021)

Most of the quantum computers currently available operate under the model of universal quantum gates. IBM, IONQ, Atom, Quantum Brilliance, IQM, and Honeywell are some of the manufacturers that offer these devices. These machines have a set number of qubits, upon which multiple gates can be applied in succession. This produces the evolution of the quantum system, which is composed of a set of qubits. At the end of this process, it is expected that the system will be in a state so the solution to the optimization problem at hand has the maximum probability of been the result of a measure. It is important to notice that even though the state of a quantum system is a superposition of all possible states, measurements collapse the wave function and leading to only one state. The measurement result is random so, even when a state has a high probability, of been the result of a measurement, the process must be repeated several times. Currently, the main challenge of the universal quantum gate model is maintaining qubit stability due to noise—a problem that intensifies as the number of qubits increases. (Sotelo, 2021)

**Identity gate:**

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

**Not gate:**

$$NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

**Hadamard gate:**

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Hadamard gate is important because when we applied to a state “0”, its output is a state of homogeneous superposition of states “0” and “1” according to Sotelo.

### 2.3.2 Universal quantum gate model

These systems operate with a fixed number of qubits and permit the sequential application of various quantum gates, influencing the system's quantum state over time. Ideally, this sequential operation prepares the system to yield a solution to a specific problem when is measured. One significant obstacle in utilizing the universal quantum gate framework is ensuring qubit stability, an issue that becomes more challenging as the qubit count escalates. Currently, this technology is in the phase known as noisy intermediate-scale quantum (NISQ), characterized by limited qubit numbers, restricted application scope, and the employment of hybrid algorithms. (Qiskit, 2022)

IBM's Q Experience and the Qiskit software framework primarily focus on universal gate-model quantum computing. In a gate-model quantum computer, qubits are manipulated using a sequence of quantum gates, and this model is more general-purpose in nature. It can be programmed to perform a wide variety of computations, from optimization and simulation to cryptography and machine learning.

IBM has a rich ecosystem around Qiskit, which includes tools and libraries designed for various quantum algorithms and applications, including optimization. In this study, this model was applied.

## 2. 4 Formulation of the portfolio problem

According to a definition put forth by Palmer (2021), the QUBO (Quadratic Unconstrained Binary Optimization) model is specifically designed for facing QUBO issues. Formally, the QUBO model is represented as an optimization problem with the equation:

$$y = x'Qx \qquad (9)$$

where x is a binary decision variable vector, and Q is a square matrix of constants.

This QUBO approach offers a mathematical framework for encoding optimization problems in a format suitable for quantum computation. While the Ising modeling technique also exists, it is not the focus of this document.

When it comes to portfolio optimization, which aims to maximize returns and minimize investment risk, the computational cost is significantly high. If investments are carried out in discrete units, the problem becomes NP-hard, further complicating matters when real-world market constraints, like transaction costs, are included.

Portfolio optimization is among the earliest and most extensively studied applications in the field of quantum computing. Recent calculations, cited by Kuchkovsky (2022), validate the promising outcomes achievable through this method when using quantum computing.

In a QUBO context, the term "unconstrained" means that the methodology does not allow constraints. To meet constraints, they must be added as penalties to the objective function. This means the method can still find solutions even if they break some of the original constraints, thus allowing for a broader exploration of potential solutions. The result includes feasible and non-feasible solutions. A feasible solution is that one that met the constraints of the problem, for example, the budget defined to calculate the optimal portfolio. This is an important point as mentioned in the results of this study.

The term "quadratic" denotes that the mathematical form of a QUBO can include either linear or paired variables. Furthermore, Kuchkovsky (2022) highlights that the QUBO model enables quantum computers to find solutions that may not strictly adhere to all constraints, thereby adding value in applications that benefit from multiple solution scenarios, such as choosing the best investment portfolio.

### 2.4.1 Quadratic Unconstraint Binary Optimization (QUBO)

Quantum and hybrid algorithms require the expression of the optimization problem in the form of a Quadratic Unconstrained Binary Optimization (QUBO) problem, defined over N binary variables. In this context, N represents the number of assets utilized in the optimization problem.

In the research conducted by Dehn (2023), a general cost function is articulated as follows:

$$F(x) = \sum_{i,j}^{N} F_{ij} x_i x_j + \sum_{i,j}^{N} f_i x_i \qquad (10)$$

Where the matrix $F_{ij} \in \mathbb{R}^{N \times N}$ is symmetric, the vector $f_i \in \mathbb{R}^N$, N binary variables $x = \{x_1, x_2, \dots, x_N\} \in \{0,1\}^N$, and $x_i = x_i^2$ for binary variables.

The solution to an optimization problem expressed in the QUBO form, is a vector, denoted $x_{\text{opt}}$, that minimizes the aforementioned cost function (10). Within the scope of portfolio optimization, the matrix $F_{ij}$ is identified as the covariance matrix of the stock returns, symbolized as $\sigma_{ij}$, while the vector $f_i$ denotes the expected return $\mu_i$. The binary variables $x_i$ represent the portfolio weights, assigned a value of 1 or 0, depending upon whether the stock is selected for the portfolio. An essential parameter, $q$, restrained to the interval [0,1], may be selected to articulate the investor's risk preference. Consequently, the cost function to be minimized is expressed as:

$$F_c(x) = q \sum_{i,j}^{N} x_i x_j \sigma_{ij} - (1 - q) + \sum_{i=1}^{N} x_i \mu_i \qquad (11)$$

Another essential parameter in this context is the budget, denoted $B$, which serves as a constraint in the optimization problem and can be expressed mathematically as:

$$\sum_{i=1}^{N} x_i \leq B \qquad (12)$$

This budget represents the number of assets selected from the available N assets to be included in the portfolio. To address this constraint, a penalty term, symbolized by $A$, is incorporated into the cost function (Dehn, 2023). A portfolio is designated as "feasible" exclusively when it adheres to the budget constraint.

In certain optimization problems, a parameter $A$ is introduced as a penalty term. Consequently, the cost function, when articulated in QUBO terms, is derived as:

$$F(x) = F_c(x) + A(B - \sum_{i=1}^{N} x_i)^2 \qquad (13)$$

## 2.5 Quantum Approximate Optimization Algorithm (QAOA)

According to the definition by Rigetti in Quantum Computing (2016), the Quantum Approximate Optimization Algorithm (QAOA) is an algorithm that finds acceptable solutions to optimization problems in polynomial time. It solves each instance of the optimization problem with generally acceptable quality in a reasonable amount of computational time.

Complementing this definition with that of Brandhofer (2022), QAOA operates using relatively short quantum circuits. This is an advantage compared to other quantum algorithms like Shor's or HHL, which require much longer circuits and, therefore, more qubits with error correction. One of the disadvantages of using QAOA is that the algorithm also involves numerical classical optimization of the variational circuit parameters, which is itself an NP-hard problem. Therefore, the effectiveness of QAOA largely depends on how well this classical optimization is carried out. More efficient heuristic strategies for the classical optimization of these parameters could significantly change the performance of QAOA, for instance the Warm QAOA that is an even more powerful tool for portfolio optimization under some restrictions.

The Quantum Approximate Optimization Algorithm (QAOA) is particularly prominent in the noisy intermediate-scale quantum (NISQ) era, a term coined by John Preskill in 2018 to describe the current state of quantum computing. In this era, quantum computers are not yet fault-tolerant but have enough qubits to start doing useful work. QAOA is one of the leading algorithms suitable for NISQ machines.

Developed by Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, QAOA is designed to solve combinatorial optimization problems. It is worth noting that while QAOA is polynomial time, the quality of the solution is generally approximate. One interesting aspect of QAOA is that it has a variable depth $p$, meaning that you can adjust the complexity of the quantum circuit used based on the resources available, essentially trading off accuracy for computational time.

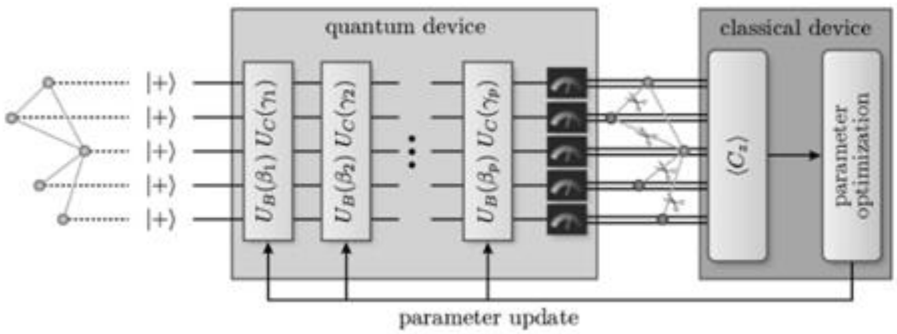


*Figure 1 Quantum Optimization Algorithms*

*Source: Capturing Symmetries of Quantum Optimization Algorithms Using Graph Neural Networks*

The classical optimization step typically uses algorithms like the Conjugate Gradient Method to optimize the variational parameters. However, the classical optimization can itself be a bottleneck, especially when the number of variational parameters is large. In practical applications, QAOA has been used in cases ranging from logistics and supply chain optimization to machine learning and financial modeling. Portfolio optimization is indeed one of the areas where QAOA could potentially offer advantages over classical algorithms, especially as quantum hardware continues to improve. (Brandhofer, 2022)

QAOA is also often compared to other quantum algorithms for optimization, such as Quantum Annealing and the Variational Quantum Eigensolver (VQE), in terms of their effectiveness and suitability for specific kinds of problems. It is an area of active research, with multiple studies exploring how to make QAOA more effective through better classical optimization techniques, hardware improvements, or even combining it with machine learning algorithms to predict good variational parameters.

Regarding to Qiskit documentation (2021), to address an optimization problem utilizing a quantum computer, it is imperative to represent the QUBO cost function through a cost Hamiltonian, denoted as $\hat{F}$. This transposition is facilitated by converting binary variables into operators, defined as $x_i = (\hat{I} + \frac{\hat{Z}}{2})$ where $\hat{I}$ and $\hat{Z}$ are the identity and the Pauli-$\hat{Z}$ operator, respectively, each acting on a qubit $i$ with $i$ ranging from 1 to $N$. Subsequently, the Quantum Approximate Optimization Algorithm (QAOA) circuit adeptly synthesizes the parameterized variational quantum state by:

$$|\psi_{\gamma,\beta}\rangle_{\text{std}} = \widehat{U}_{\text{std}}(\beta p) - e^{-i\gamma p\hat{F}} \ldots \widehat{U} std(\beta 1) e^{-i\gamma 1 p\hat{F}} \, |\psi 0\rangle \, std \quad (14)$$

with parameters $\vec{\gamma}$ = (γ1, ..., γp), $\vec{\beta}$ = (β1, ..., βp) and number of iterations p, and standard mixer $\widehat{U}$std:

$$\widehat{U}\text{std}(\beta) = e^{i\beta} \sum_{i=1}^{N} \hat{x} \quad (15)$$

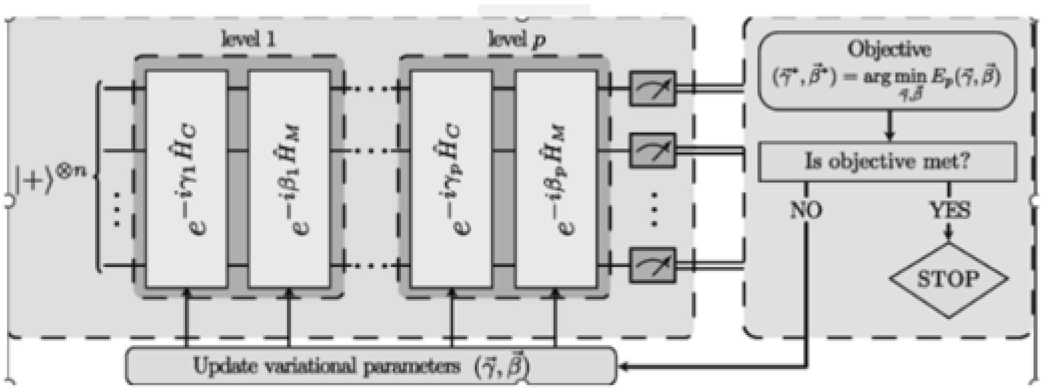


*Figure 2 Quantum Approximate Optimization Algorithm*

*Source: Applying the Quantum Approximate Optimization Algorithm to the Tail-Assignment Problem*

The initial state is carefully selected as $|\psi 0\rangle = |+\rangle^{\otimes N}$, representing the minimum energy eigenstate of the mixing operator $-\sum \hat{x}$. Following this, all qubits undergo into a measurement in the computational basis to ascertain the expectation value $\langle \hat{F} \rangle$. This intermediate result is subsequently relayed to a classical optimizer, which adeptly updates the parameters with the objective of minimizing the expectation value.

To update the variational parameters, the initial randomly parameters γ and β are passed to a classical optimization routine, which proposes new values for the parameters γ and β in order to minimize the expected value of the cost function. (Qiskit, 2021)

**2.5.1 From QUBO to Hamiltonian**

The goal of this process it to find a Hamiltonian operator $H_F$ that encodes the cost function $F(X)$. A Hamiltonian Operator corresponds to the total energy of a quantum system described by a Hermitian matrix $H = H^{\dagger}$. (Qiskit, 2021)

Following the Qiskit documentation, the energy level of a system in state $|\psi\rangle$ is given by an expectation value $E|\psi\rangle = \langle\psi|H|\psi\rangle$. And the Ground State is the lowest energy state $|\psi^*\rangle$ of a quantum system.

$$|\psi^*\rangle = \arg\min_{|\psi\rangle\in\mathcal{H}} E\,|\psi\rangle$$

Encoding the cost function into a Hamiltonian operator $H_F$ is as follows:

$$H_F\,|x\rangle = F(x)\,|x\rangle \quad (16)$$

The QUBO cost function is as follows:

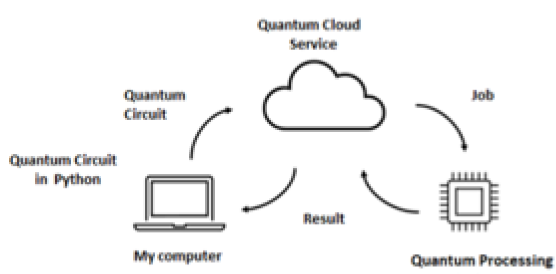


*Figure 3 Hybrid classic and quantum approach*

$$Fc(x) = \sum_{i,j=1}^{N} x_i x_j \sigma_{ij} + \sum_{i=1}^{N} x_i \mu_i = \sum_{i,j=1}^{N} x_i x_j \sigma Q_{ij} + \sum_{i=1}^{N} c_i x_i \quad (17)$$

And the Hamiltonian operator:

$$Fc = \sum_{i,j=1}^{N} \frac{1}{4} Q_{ij} Z_i Z_j - \sum_{i=1}^{N} \frac{1}{2} (c_{i+} \sum_{j=1}^{N} Q_{ij}) Z_i + (\sum_{i,j=1}^{N} \frac{Q_{ij}}{4} + \sum_{i,j=1}^{N} \frac{Qc_i}{2}) \quad (18)$$

**2.5.2 QAOA quantum circuit.**

The QAOA approach to optimization involves preparing a quantum state in superposition and applying carefully constructed layers to guide the quantum system toward a state where a measurement is likely to yield a good approximation to the optimal solution of the combinatorial problem. Parameters $\beta$ and $\gamma$ are tuned to minimize the expectation value of the cost function, often using classical optimization algorithms in a hybrid quantum-classical loop. This method is particularly noted for problems like portfolio optimization, where a combination of asset selections (binary variables) needs to be optimized against a cost function representing risk or reward. (Qiskit, 2021)

In the context of portfolio optimization, QAOA is applied to find the optimal combination of financial assets to minimize risk or maximize return, considering the combinatorial nature of selecting a subset of assets from a larger collection of stocks. The quantum circuit attempts to navigate through the solution space efficiently by leveraging superposition and carefully structured interference, via the cost and mixer layers, to enhance the probability of measuring a good solution in the computational basis. This approach is beneficial because the computational complexity of evaluating all possible portfolios becomes unmanageable with classical approaches as the number of considered assets grows. (Qiskit, 2021)

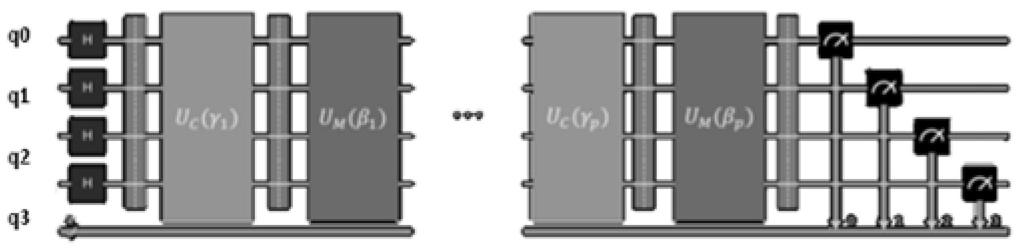


*Figure 4 Process of the QAOA. Source: Qiskit documentation.*

1) Preparation of equal superposition state:

$|+\rangle^n = \sum_{X\in\{0,1\}^n} \frac{1}{\sqrt{2^n}} |x\rangle$

2) p repetitions of alternating cost and mixer layers:

$Uc(\gamma i) = e^{-i\gamma i Fc}$

$U_M(\beta i) = e^{-i\beta i Fc}$

3) Measurement in computational basis.

The QAOA process begins with the initial state preparation, where every qubit is prepared in an equal superposition state using Hadamard gates, meaning that a Hadamard gate is applied to each qubit that starts in the $|0\rangle$ state, placing them into a superposition of $|0\rangle$ and $|1\rangle$ states. After this, the application of layers takes place: the QAOA circuit applies *p* layers, each consisting of a cost layer and a mixer layer, with the cost layers being parameterized by angles labeled $\gamma$ and the mixer layers by $\beta$. This involves the implementation of cost and mixer Hamiltonians, wherein the cost layer involves the exponentiation of the cost Hamiltonian, and the mixer layer involves the exponentiation of a mixer Hamiltonian. Upon the application of these layers, a measurement is finally performed in the computational basis. (Qiskit, 2021)

### 2. 6 Other Hybrid Algorithms

#### 2.6.1 Variational Quantum Eigensolver Algorithm.

Among the most well-known quantum algorithms, according to Peruzzo (2014), is the Variational Quantum Eigensolver (VQE). It's a hybrid quantum-classical algorithm for optimization. It involves carrying out a variational optimization of a quantum state to obtain an approximation of the ground state of a Hamiltonian. After estimating the energy of the quantum state through sampling, the parameters are finely adjusted to reduce the energy using classic methods like, for example, the conjugate gradient method or a quantum approach like the parameter shift rule (Mari, Bromley, & Killoran, 2020). After a certain number of iterations, the energy converges, producing an approximation of the desired ground state. The performance of VQE strongly depends on several aspects, but the most important is the choice of the variational quantum circuit.

VQE is especially useful when dealing with systems that are too large to be accurately simulated on a classical computer but small enough to be manageable on a current quantum computer. Moreover, being a hybrid algorithm, VQE benefits from the advantages of both classical computational resources and quantum ones. In this study, exclusive utilization of the Quantum Approximate Optimization Algorithm (QAOA) is adopted, given its status as a specialized instance of VQE. (Peruzzo,2014)

### 2.7 Qiskit

Developed by IBM, Qiskit is an open-source quantum software development kit. It has multiple libraries offering to different layers of quantum computation, ranging from foundational elements to quantum machine learning. Users can also access IBM's real quantum processors via IBM's Quantum Cloud services to run their programs. The platform is continually updated, incorporating both academic and industrial research findings to stay on the edge of quantum advancements.

#### 2.7.1 Qiskit Aer

Qiskit Aer is one of the four elements of Qiskit, designed to work as a simulator backend for Qiskit's software stack. The "Aer" stands for "A Class of Exact Solvers," emphasizing its utility in simulating complex quantum circuits in classic computers. The simulation in a classic computer can include different types of noise, allowing for more realistic modeling of how a quantum algorithm would behave on an actual quantum computer. By understanding how noise impacts algorithm performance, researchers can explore error mitigation techniques and improve algorithm robustness.

Portfolio Optimization with QAOA is particularly promising for tackling complex combinatorial optimization problems like portfolio optimization. QAOA can find 'good enough' solutions within a polynomial time frame, making it a practical tool for real-world applications. When using Qiskit Aer for such applications, users can gain insights into how their algorithms will perform under realistic conditions, making it easier to fine-tune and optimize them for real quantum hardware. This is invaluable in a field like financial portfolio optimization where small improvements can lead to significant financial gains. (Qiskit, 2021)

### 2.8 Quantum Lab.

In the context of this study users are afforded the capability to compose scripts that integrate Qiskit code and mathematical equations within a Jupyter Notebook environment, obviating the necessity for any installation. The platform enables the execution of code on authentic quantum hardware or, alternatively, on simulators, providing the additional utility of storing, accessing, and managing files from any geographical location.

The Quantum Approximate Optimization Algorithm (QAOA) is indeed a powerful tool for solving combinatorial optimization problems for exploring feasible solutions to complex optimization problems, and it can be implemented using Python-based quantum programming frameworks and run on local computers or on cloud-based quantum computing platforms like IBM Quantum Lab.

To work with QAOA, it is possible to use popular Python-based quantum programming frameworks such as Qiskit, that offer a convenient interface for programming quantum circuits and running quantum simulations on classical computers or actual quantum hardware.

QAOA can be implemented in Python code and executed on a local computer equipped with a compatible quantum simulator to simulate the behavior of quantum circuits on classical hardware, enabling the development and testing of quantum algorithms without the need for quantum hardware. Alternatively, there is access to cloud-based quantum computing platforms like IBM Quantum Lab, which provide remote access to real quantum processors and quantum simulators over the internet. (Qiskit, 2021)

In summary, QAOA, provides a valuable tool for exploring optimization problems with datasets and constraints. It allows researchers and developers to investigate the solution landscape of complex optimization problems, helping an investor to check different portfolio options that meet a set of financial constraints.

In this study, the basic version of QAOA in Qiskit is utilized for portfolio optimization, which is readily available for use. It is also possible to adjust parameters such as the circuit depth, iterations, and constraints of the optimization problem as the number of assets, the risk aversion, the budget and number of assets to assess their impact on performance.

## 3. Methodology

The methodology is organized in a clear and sequential manner as outlined below:

**3. 1 Section 1:** High-correlation stock analysis and hybrid methodology. Initially, a traditional approach to portfolio optimization is applied, specifically employing the Markowitz and Black-Litterman models on a subset of 10 assets, all of which exhibit high correlation.

The second phase involves deploying our hybrid classical-quantum methodology, leveraging the Quantum Approximate Optimization Algorithm (QAOA) on the dataset of the same 10 stocks. The objective centers around selecting an optimal subset of 5 out of these 10 stocks. Computations during this phase are executed using Google Colab. In a final step within this subsection, the identical methodological approach is applied, but with an altered focus on choosing an optimal 10 out of a wider pool of 20 assets from the SP500, employing the Quantum Lab cloud-based platform for computational processes.

*Table 1 Simulations scenarios 1*

| Simulation ID | Methodology | Assets Analyzed | Objective | Simulation Environment |
|---|---|---|---|---|
| Simulation 1 | | | | |
| Simulation 1.1 | Markowitz and Black-Litterman | 10 first assets from SP500 | Find optimal weights with W $\in (-\infty,+\infty)$ | Google Colab |
| Simulation 1.2 | Markowitz and Black-Litterman | 10 first assets from SP500 | Find optimal weights $w > 0$ | Google Colab |
| Simulation 2 | QAOA | 10 first assets from SP500 | Select 5 best assets to optimize a portfolio | Google Colab |
| Simulation 3 | QAOA | 20 first assets from SP500 | Select 10 best assets to optimize a portfolio | Google Colab, Quantum Lab IBM |

**3.2 Section 2**: Low-correlation stock analysis and comparative assessment. This section replicates the simulation from the preceding section 1, although with an altered focus on the 20 least correlated stocks from the SP500. An analytical examination follows, scrutinizing and contrasting the performance outcomes of both traditional and hybrid methods to optimize portfolios. These results are subsequently compared with the findings detailed in Section 1, ensuring a comprehensive comparative analysis.

*Table 2 Table 3 Simulations scenarios 2*

| Simulation ID | Methodology | Assets Analyzed | Objective | Simulation Environment |
|---|---|---|---|---|
| Simulation 4 | | | | |
| Simulation 4.1 | Markowitz and Black-Litterman | 10 less correlated assets from SP500 | Find optimal weights with W ∈ (−∞,+∞) | Google Colab |
| Simulation 4.2 | Markowitz and Black-Litterman | 10 less correlated assets from SP500 | Find optimal weights with w > 0 | Google Colab |
| Simulation 5 | QAOA | 10 less correlated assets from SP500 | Select 5 best assets to optimize a portfolio | Google Colab |

It is important to mention that QAOA results are organized into optimal solutions, feasible but non-optimal solutions, and non-feasible solutions, consisting of hundreds of binary vectors with their respective Fvale and probabilities. For this study, only the optimal solution and the first 9 feasible solutions are shown for comparison with the optimal solution.

For all simulations, the hyperparameters to be utilized in the ready-to-use algorithms are detailed as:

n: Number of assets.
μ defines the expected returns for the assets.
σ specifies the covariances between the assets.
q: Risk factor or aversion.
p: depth of the quamtun circuit.
Budget: Number of assets/2.
Penalty term: Equal to the number of assets, used to scale the budget penalty term.
Maxiter: number of iterations of the quatum circuit.

The tables below, corresponding to the simulation items, specify the exact values of the hyperparameters employed.

In the context of portfolio optimization, the next hyperparameters are important because they guide the algorithm of Markowitz, Black-Litterman and QAOA in exploring the solution space efficiently and can significantly affect the outcome and performance of the optimization process.

1. n (Number of assets): This defines the size of the cost function in the optimization problem. The number of assets sets the dimensionality of the problem space, impacting the complexity of the optimization process. A higher number of assets increases the search space for the optimal portfolio, which may make the problem more computationally demanding.

2. µ (Expected returns for the assets): The expected return for each asset is a fundamental input in determining the attractiveness of an investment. Higher expected returns come with higher risk.

3. σ (Covariances between the assets): The covariance matrix is essential for understanding the relationships between the returns of different assets. Diversification, which is an essential part of risk management, is achieved by selecting assets with lower covariances, thereby reducing the overall portfolio risk.

4. q (Risk factor or aversion): The risk aversion hyperparameter measures the investor's willingness to take on a risk. This hyperparameter helps in aligning the optimization process with the investor's risk profile.

5. p (Depth of the quantum circuit): In the QAOA method, the depth of the circuit is the complexity of quantum operations required to solve the problem. The depth of the circuit affects the ability of the quantum algorithm to explore the solution space and can impact the accuracy and precision of the results.

6. Budget (Number of assets/2): The budget constraint typically sets a limit on the number of assets that can be included in the optimal portfolio, in this case, half the number of available assets.

7. Penalty term (Equal to the number of assets): The penalty term is used to enforce the budget constraint within the optimization process. With penalty term, the algorithm penalizes (non-feasible) solutions that does not meet the budget constraint.

8. Maxiter (Number of iterations of the quantum circuit): This factor sets the maximum number of iterations that the QAOA algorithm will run. Too many iterations can lead to a more accurate solution but will take more computational time.

In summary, these hyperparameters play essential roles in guiding the portfolio optimization process. They need to be carefully selected and tuned to reflect the investor's objectives and constraints, the characteristics of the financial assets, and the capabilities and limitations of the quantum computing framework being used.

**3.3 Section 3**: Backtesting and forward-looking perspectives in the concluding section, results derived from prior analyses are subjected to backtesting to affirm their validity and reliability. Conclusions are articulated based on these backtesting outcomes, and the section culminates with a brief insightful perspective, interpreting potential possibilities for future investigations in the area of portfolio optimization.

**4. Description of the data sets.**

The data set used in this study comes from the Standard & Poor's 500 (S&P 500) index, one of the most representative and observed stock market indices in the world. Composed of 500 of the largest companies in the United States, the S&P 500 is considered a reliable indicator of the performance of the U.S. stock market and, by extension, of the country's economy. The relevance of the S&P 500 is due to its ability to serve as an economic and financial barometer. The index covers a wide range of sectors, making it a natural tool for diversification and, therefore, an ideal reference point for investment strategies and academic studies. This data was collected using the "yfinance" library to download the data directly from Python.

*Table 3 Datasets used in Markowitz, Black-Litterman and QAOA optimizations.*

| Dataset | Description |
|---|---|
| 1 | The first 10 stocks from the SP500 (with high correlation). |
| 2 | The first 20 stocks from the SP500 (with high correlation). |
| 3 | The 10 stocks from the SP500 with the lowest correlation. |

All datasets are industry diversified by choosing assets from different industry sectors.

- First ten assets of SP500: ["ABBV","ABT","ADBE","ADM","AES","ALB","ALE","AOS","APD","MMM"]

- First twenty assets of SP500: ['A', 'ABBV', 'ABNB', 'ABT', 'ACN', 'ADBE', 'ADM', 'ADP', 'AES', 'AFL', 'AKAM', 'ALB', 'ALGN', 'ALK', 'ALLE', 'AOS', 'APD', 'ARE', 'ATVI', 'MMM']

- Ten assets with lowest correlation: ["MRNA","NEM","KR","CLX","CPB","PCG","SJM","ENPH","BIIB","EQT"]

*Table 4 Characteristics of the datasets each set includes the following variables for each of the assets that make up the S&P 500*

| Variable | Description |
|---|---|
| Opening | The opening price is the price at which the stock began trading during a specific period, most commonly when the trading day begins |
| High | The highest price represents the maximum trading value of the stock during that specific period. It indicates the peak point where the stock was traded. |
| Low | This is the lowest price at which the stock was traded during the specific period. It indicates the minimum valuation that investors were willing to pay for the stock during that time. |
| Close | The closing price is the final trading value of the stock at the end of the trading period, which often coincides with the stock market's closing for the day. |
| Adjusted Close | This is a more nuanced version of the closing price, adjusted for events such as dividends, stock splits, and new stock offerings. It provides a more accurate representation of the stock's value over time. |
| Volume | This represents the total number of shares traded during a specific period. A high trading volume may indicate strong interest in a stock, while low volume could suggest a lack of liquidity or interest. |

Source: Wikipedia, 2023

While the aforementioned variables are important for statistical purposes and for comprehensively understanding stock behaviors due to their ability to provide a thorough view of the behavior of each asset, this study used 'Adj Close' for the optimization problem. Utilizing 'Adj Close' as opposed to other price metrics like 'Close,' 'Open,' 'High,' or 'Low' has several advantages in this context. 'Adj Close' is adjusted to account for events such as dividends paid to shareholders and stock splits, which is vital for calculating returns over time since a dividend or stock split can significantly distort the perception of an asset's performance compared to the closing price ('Close'). Furthermore, the adjusted closing price reflects the actual value an investor would realize if they had held a stock over a period. In other words, it considers not merely fluctuations in the stock price but also additional income in the form of dividends, specifically for the period from January 1, 2014, to January 1, 2023.

### 4.1 Data preprocessing prior to portfolio optimization

Ensuring data integrity and relevance is crucial in portfolio optimization applications. The datasets utilized in this study underwent a series of preprocessing steps to refine the data, ensuring its viability for subsequent optimization techniques. The datasets underwent preprocessing steps as follows:

#### 4.1.2 Logarithmic return computation

The basis for any portfolio optimization technique is the asset's return over time. In line with modern financial theory, logarithmic returns were calculated due to their advantageous mathematical properties including additivity and symmetry. This was achieved via Python's pct_change() function to compute inter-row percentage changes, followed by application of np.log(1 + x) to each element to yield the logarithmic return.

### 4.1.3 Imputation of missing values

Financial time-series data often suffer from missing entries owing to various corporate actions, such as stock splits or mergers. To handle this issue and maintain data integrity, missing values were imputed using the fillna() function in Python. This function replaced missing entries with the columnar mean, thereby preserving the underlying statistical properties of the dataset.

### 4. 1.4 Data Cleansing via null-value elimination

After the imputation phase, any lingering rows that contained NaN (Not a Number) values were cleansed from the dataset to ensure computational reliability and robustness. This was executed via Python's dropna() function.

Upon successful execution of the above-mentioned preprocessing tasks, the dataset attained a level of maturity and reliability that performed it fit for subsequent portfolio optimization algorithms.

## 5. Results

### 5.1 Section 1

The selected assets for this section are the 10 first stocks belonging to companies from different sectors of the economy, ranging from healthcare and technology to industrial goods and services. According to the S&P 500 index listing (Wikipedia, 2023).

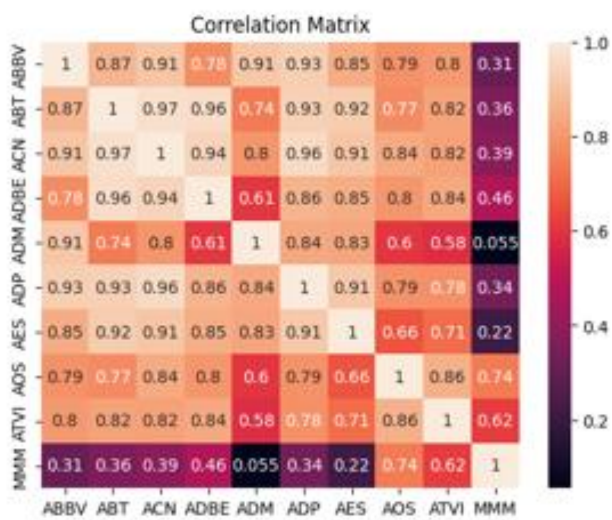


*Figure 5 Correlation matrix of the dataset 1. Ten assets with high correlation*

The correlation matrix provides information about the linear relationship between the different stocks or assets in the portfolio.

Highly correlated assets: If two assets have a correlation coefficient close to 1, it means they tend to move in the same direction. This could be risky for portfolio diversification, as a drop in one could imply a drop in the other.

Low correlation assets: A correlation coefficient close to 0 indicates that the assets are independent of each other in terms of their price movement. This could be a good opportunity for risk diversification.

Negatively correlated assets: A correlation coefficient close to -1 indicates that the assets move in opposite directions. If one asset goes up, the other tends to go down, and vice versa. This can also be useful for diversification.

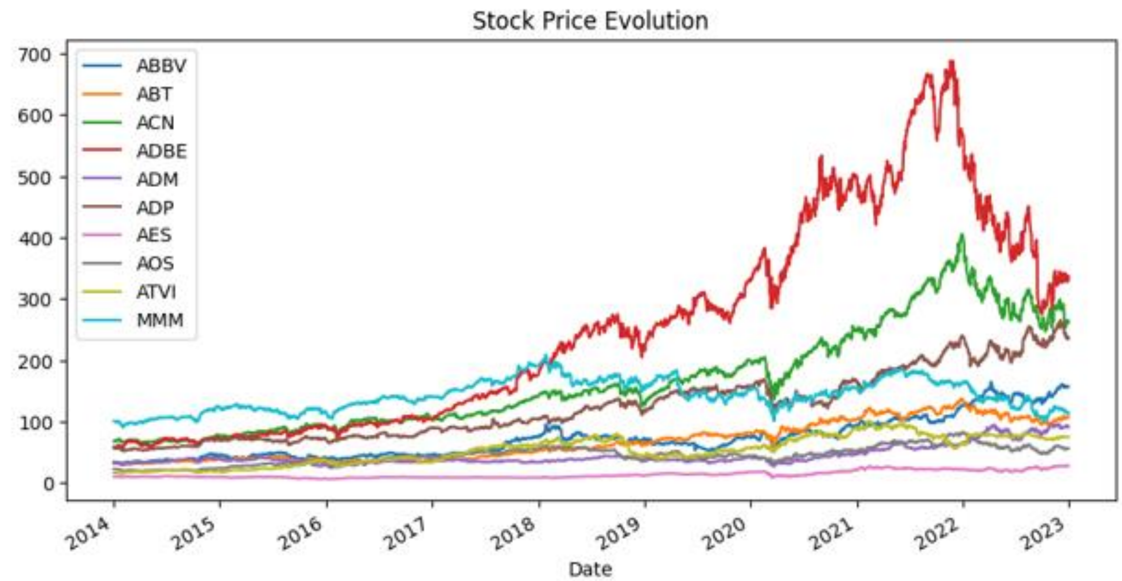


*Figure 6 Stock price evolution from 2014 to 2023.*

**5.1.2. Simulation 1**

The Markowitz method allowed us to perform multiple quadratic optimizations for different target returns, resulting in the identification of the portfolios that make up the efficient frontier. These portfolios represent the optimal combinations of assets in terms of performance and risk, given the specified constraints. The key steps of the method are as follows:

**Expected returns:** We calculated the expected returns for all assets in the dataset, using historical information and statistical techniques.

**Covariance matrix:** A covariance matrix was prepared to understand how the returns of different assets correlated with each other. This step is crucial for evaluating the risk associated with a combination of different assets.

**Optimization:** Quadratic programming was employed to solve an optimization problem aimed at minimizing a function composed of the portfolio variance and the expected return. This optimization was subject to various constraints, including the condition that the sum of the weights assigned to each asset in the portfolio must equal one.

**5.1.3 Different cases of studies**

To find the portfolio optimization problem using Markowitz and Black Litterman, we engaged in an optimization exercise consistent with the principles of modern portfolio theory. To start, we established some of the fundamental variables for our analysis, such as the number of assets N, expected returns: $\mu$, and the covariance matrix $\Sigma$, all calculated from preprocessed historical data.

The next step involved defining a variable w to represent the distribution of portfolio weights among the assets. We then calculated the portfolio's expected return between the assets' expected returns $\mu$ and their respective weights w. Simultaneously, we calculated the portfolio's risk, represented by its variance, using the quadratic form of the weights w and the covariance matrix Σ. Our optimization problem included different essential constraints as follows:

*Table 5 Constraints used in Markowtiz and Black Litterman optimization.*

| Constraints | Simulation 1.1 | Simulation 1.2 |
|---|---|---|
| Target return | 0,05 | Expected return |
| $\sum wi$ | 1 | 1 |
| Wi | W ∈ (−∞,+∞) | wi>0 |
| Assets | 10 | 10 |

The portfolio optimization process involves several constraints to shape the risk and return profile of the investment portfolio. Here, we applied four critical constraints:

- Minimum expected return constraint: This constraint mandates that the portfolio's expected return should be at least equal to a predetermined target return. This serves to align the portfolio construction with the investor's return expectations or requirements.
- Fully invested portfolio constraint: The second constraint ensures that the sum of the portfolio weights equals one. This is essential to assure that the entire capital is allocated, resulting in a fully invested portfolio without any uninvested cash.
- Allowance of short positions constraint permits the portfolio weights to take on both positive and negative values, thereby allowing for the incorporation of short selling of assets. It is crucial to note that this case is simulated solely to comprehend the outcomes of the Markowitz and Black-Litterman optimization methods under negative weights. However, it will not be employed in the remaining facets of the study, as scenarios involving short selling are not within the scope of this paper.
- Asset number constraint: Lastly, the fourth constraint concerns to the limitation in the number of assets included in the portfolio. For clear visualization and to avoid portfolio clutter, we select the first ten assets. This methodological choice is crucial to prevent overwhelming diversification and to maintain focus on a select group of stocks.

**5.1.4 Results of Markowitz optimization simulations 1.1 and 1.2.**

*Table 6 Markowitz weights simulation 1.1 and 1.2*

| Stock | Wi simulation 1.1 | Wi simulation 1.2 |
|---|---|---|
| ABBV | 18,35 | 0,142 |
| ABT | 2,04 | 0,152 |
| ACN | 16,46 | 0,015 |
| ADBE | 11,19 | 0 |
| ADM | 4,7 | 0,16 |
| ADP | 30,88 | 0,064 |
| AES | -7,04 | 0,047 |
| AOS | 3,67 | 0,077 |
| ATVI | 4,73 | 0,124 |
| MMM | -83,96 | 0,218 |

**5.1.5 Results of Black Litterman optimization simulations 1.1 and 1.2.**

In the Black-Litterman model, views and view uncertainties serve as central elements in adjusting market equilibrium returns to derive the posterior expected returns, which are subsequently utilized to optimize the portfolio. A views array embodies the investor's expectations or perspectives on the returns of the assets within the portfolio.

Here, the views were selected hypothetically as follows:

views = np.array([0.05, 0.07, -0.03] + [0.0] * (n - 3))

This implies that the investor has specific return expectations for the first three assets: 5%, 7%, and -3%, respectively. For the remaining (n - 3) assets, no specific views are expressed, and they are assigned a return expectation of 0.0. In essence, the investor anticipates positive returns for the first two assets, expects the third asset to decline, and has no strong opinion on the remaining assets. In real-world scenarios, the Black-Litterman method depends on the investor's skill in selecting views and can, in some cases, be subjective.

**View Uncertainty:**
View uncertainty, represented by Omega, quantifies the confidence in the investor's views. It's defined as:

view_uncertainty = np.eye(n) * 0.1

This denotes that the uncertainty associated with each view is represented as a diagonal matrix where each entry is 0.1. This choice signifies a uniform confidence level associated with each expressed view.

These views and uncertainties are combined with the market equilibrium returns (Pi) and the tau-scaled inverse covariance matrix of asset returns to calculate the posterior expected returns. The Black-Litterman model integrates investor's views and the market equilibrium to produce a set of returns that reflect both market expectations and investor's beliefs, which are subsequently used to construct the optimal portfolio.

*Table 7 Black-Litterman weights simulation 1.1 and 1.2*

| Stock | Wi simulation 1.1 | Wi simulation 1.2 |
|---|---|---|
| ABBV | 18,5 | 0,142 |
| ABT | 2,02 | 0,15 |
| ACN | 16,34 | 0,019 |
| ADBE | 11,81 | 0 |
| ADM | 4,46 | 0,159 |
| ADP | 30,4 | 0,066 |
| AES | -7,04 | 0,048 |
| AOS | 3,06 | 0,079 |
| ATVI | 4,91 | 0,123 |
| MMM | -83,46 | 0,214 |

We had in both, the Markowitz optimization model and the Black-Litterman model are yielding similar portfolio weights, this could suggest a few things:

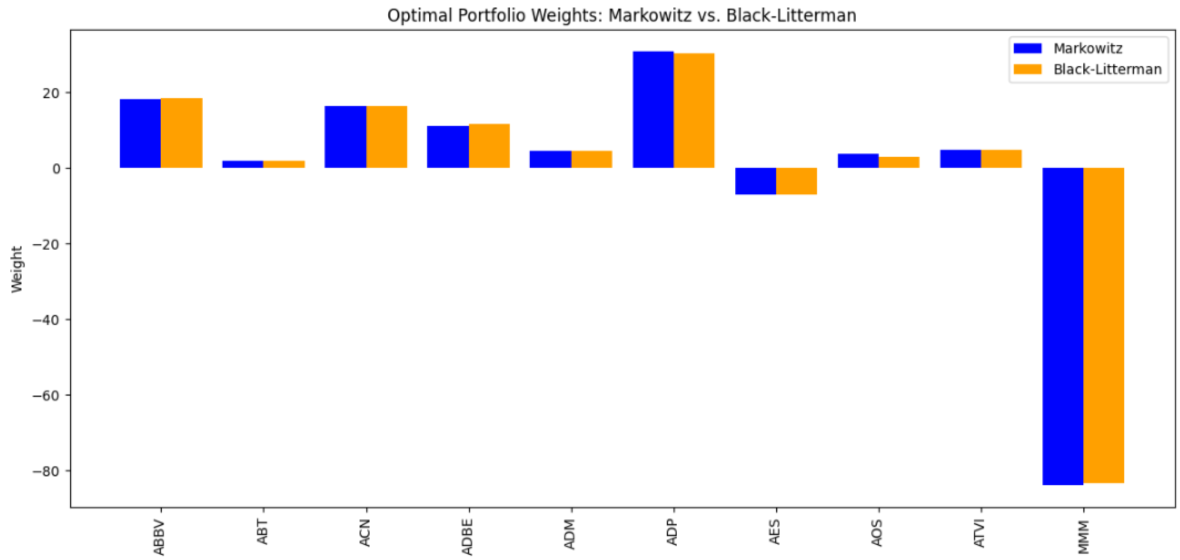


*Figure 7 Optimal portfolio weights simulation 1.1: Markowitz vs Black-Litterman.*

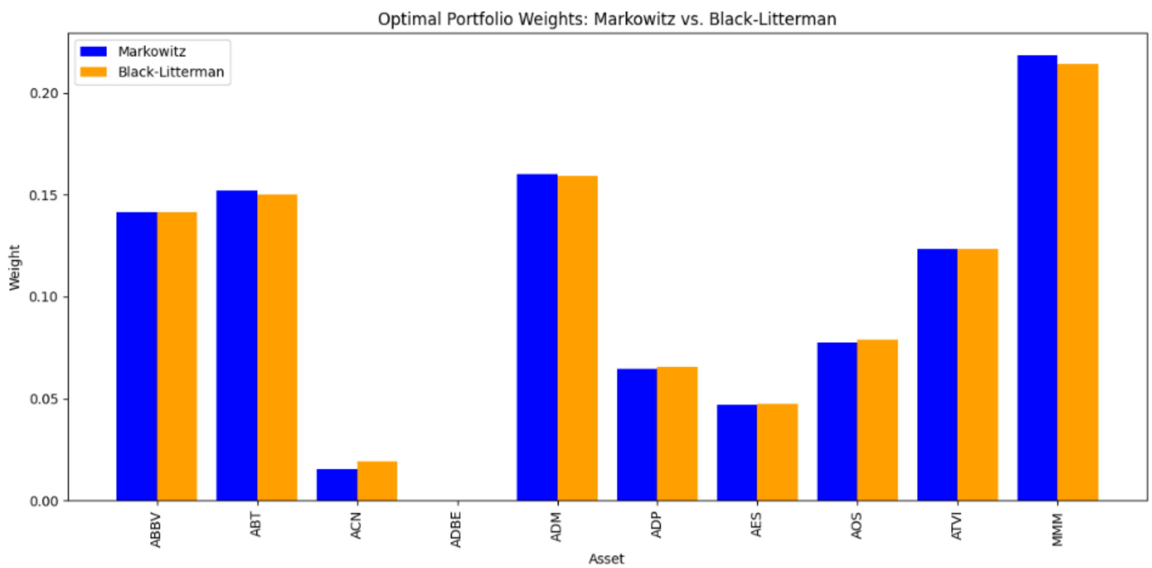


*Figure 8 Optimal portfolio weights simulation 1.2: Markowitz vs Black-Litterman.*

**Consistent views and market equilibrium:** The investor's views, as represented in the Black-Litterman model, may be quite consistent with the implied market equilibrium returns. If the investor's views are not significantly different from the market equilibrium, then the Black-Litterman model would naturally converge to a solution like the traditional Markowitz model.

**Neutral Views:** If the views applied in the Black-Litterman model are relatively neutral or are assigned with high uncertainty, this could cause the model to lean more towards the market equilibrium, thereby aligning more closely with the Markowitz model's outputs.

**5.2 Interpretation of the results.**

To analyze the risk-return trade-off, a portfolio was constructed from a subset of 10 stocks within the S&P 500 index. Utilizing historical data, we established a set of constraints, which, in some cases, included a target expected annual return of at least 5% and an expected return approximately equal to the median. For the purposes of this study, negative weights will not be considered. Regarding the results of simulation 1.2, we have:

**Expected return as the mean of the assets**: Aiming for a portfolio expected return equal to the average return of the constituent assets is a common approach, especially if the investor wants a performance that, at least, tracks the average market return (assuming the assets are representative of the market). Depending on the individual returns and volatilities of the assets, the optimization will seek a weight distribution that achieves this return with the minimum possible risk, given the other constraints.

**Non-negative weights constraint:** Ensuring all weights are non-negative restricts the optimization to long-only strategies, prohibiting short-selling. This is common in many portfolio optimizations as it prevents betting against any assets and ensures a more straightforward investment strategy.

Given the Markowitz and Black-Litterman weights:

- **Diversification**: The portfolio is diversified across various stocks, which is crucial for risk mitigation by avoiding over-reliance on the performance of a single asset.
- **Significant weights on certain assets**: MMM, ADM, and ABT carry the highest weights, indicating that the portfolio is most heavily invested in these stocks. These assets will notably influence the portfolio's overall performance and risk-return profile.
- **Zero allocation to ADBE**: A weight of 0 for ADBE suggests that this asset does not contribute positively to achieving the desired return-risk trade-off under the given constraints.
- **Minimal investment in ACN**: The small weight for ACN implies that increasing its allocation may not be beneficial for optimizing the portfolio under the specified conditions.

The weights ($w_i$) obtained from this method indicate the proportion of the total capital that should be allocated to each asset in the portfolio. These weights serve as a direct guide for portfolio construction. Here are some key points on how to interpret and use these weights:

**Interpreting the weights**

1. Positive weights: A positive weight for an asset indicates that one should have a "long position" in that asset. For example, in the Markowitz optimization simulation 1.1, a weight of 18.34% for the asset "ABBV" suggests that 18.34% of the portfolio's total capital should be invested in that asset.
2. Negative Weights: A negative weight indicates a "short position," meaning that the asset should be sold short. For example, in the Markowitz optimization case 1.1, a weight of -7.04% in the asset "AES" suggests it should be shorted, and the released capital could be used to take long positions in other assets.
3. Sum of weights: The total sum of the weights must equal 100% (or 1 when the weights are normalized), indicating that all capital is utilized.

**How to construct a portfolio**

1. Divide the capital: Divide the total capital according to the weights. For example, if you have a total capital of $10,000 and the weight for the asset "ABBV" is 18.34%, then invest $1,834 in "ABBV."
2. Short Positions: For assets with negative weights, take short positions and use the freed-up capital to finance long positions in other assets.

3. Commissions and fees: Consider buying and selling commissions and any other associated fees when adjusting the weights. For that reason, short positions are not suggested by investors. The incorporation of buying and selling commissions, alongside any pertinent fees, is crucial when adjusting asset weights. This consideration becomes especially vital in strategies involving frequent trading or re-balancing of assets, where the cumulative effect of commissions and fees can significantly erode the portfolio's net return. Consequently, investors often exhibit a tendency for strategies that minimize the need for short positions, given the potential for additional costs and complexities associated with them.

4. Rebalancing: The weights may change over time due to fluctuations in asset prices. It's important to regularly rebalance the portfolio so that it adheres to the optimal weights.

5. In the landscape of portfolio optimization, particularly within the frameworks of Markowitz and Black-Litterman models, the well-informed selection of constraints is a critical determinant in facilitating viable investment strategies. For instance, in the case where a constraint $w_i \geq 0$ and a high expected return were imposed, achieving a solution for the optimization problem was unattainable with certain high target expected return. This implies that under these restrictions and given the inherent characteristics and correlations of the assets involved, the portfolio could not meet the specified expected return while maintaining non-negative asset weights. This occurrence underlines the importance of comprehensively analyzing the feasibility of imposed constraints in conjunction with the desired investment outcomes, to ensure that the optimization problem remains solvable and aligns with practical investment scenarios and goals.

### 5.3 Quantum Approximate Optimization Algorithm

The implementation process of the Quantum Approximate Optimization Algorithm (QAOA) was divided into two scenarios. In the initial scenario, we selected the 10 first stocks from the S&P500 with the aim of constructing an optimal investment portfolio composed of 5 of these assets. Simulations for this stage were executed on a desktop computer and Google Colab, where both resources proved adequately proficient to manage the computational demands without issues. However, the case, in which the dataset was expanded to include the 20 first assets with the intention of discerning an optimal portfolio of 10 assets, presented a significant computational challenge that surpassed the capabilities of the desktop computer and a cloud base resource as Google Colab Pro. Consequently, we found it necessary to transition our simulations to the IBM Quantum Lab environment, which, with its robust computational capacity, facilitated an approximate solution to the complex portfolio optimization without much technical problems.

To address the portfolio optimization problem, we adopted a hybrid strategy that leverages both classical and quantum computing capabilities. This innovative approach offers the computational speed and versatility of classical algorithms, coupled with the theoretical optimality achievable through quantum algorithms. As a first step in this direction, we set up a Python environment compatible with both classical and quantum computing libraries.

*Table 8 Classical Computing Libraries*

| Library | Description |
|---|---|
| NumPy and Pandas | for numerical and data manipulation. |
| SciPy | for classical optimization routines like SLSQP (Sequential Least SQuares Programming). |

| | |
|---|---|
| Matplotlib or Seaborn | for data visualization. |

*Table 9 Quantum Computing Libraries.*

| Library | Description |
|---|---|
| from qiskit.algorithms import QAOA | Imports the Quantum Approximate Optimization Algorithm (QAOA) from the Qiskit library. This algorithm is used for solving optimization problems on a simulated or real quantum computer. |
| from qiskit_optimization.algorithms import MinimumEigenOptimizer | Imports an optimizer that solves optimization problems using the minimum eigenvalue of a matrix. |
| from qiskit_optimization.converters import QuadraticProgramToQubo | QuadraticProgramToQubo Imports a class that converts quadratic programming problems to QUBO (Quadratic Unconstrained Binary Optimization) problems, which are easier to solve with simulated or real quantum computers. |
| from qiskit.algorithms.optimizers import COBYLA | Imports the COBYLA (Constrained Optimization BY Linear Approximations) optimizer, which is an algorithm for numerical optimization. |

### 5.3.1 Simulation 2.

**Algorithm parameters and simplifying assumptions.**

To perform portfolio optimization using hybrid quantum algorithms, we initially defined a series of parameters and assumptions. The parameters include:

*Table 10 Parameters for QAOA simulation 2*

| Parameter | Value |
|---|---|
| Number of Assets (n) | 10 |
| Risk Factor (q) | 0.5 |
| Budget: (Number of Assets/2) | 5 |
| Penalty: Equal to the number of assets, used to scale the budget penalty term. | 10 |

The risk factor, denoted as q, plays a fundamental role as a moderator between return and risk within the objective function, ensuring a balanced approach to investment strategy. Simultaneously, the budget, defined as half the total number of assets (number of assets/2), regarding to a study conducted by Gomez (2022) which indicates that the QAOA method performs best when the constraint budget is set at half the number of assets. This budget dictates the necessary inclusions in the optimized portfolio, conducting the allocation direction. Increasing the number of assets significantly increases the complexity of identifying a solution within polynomial time, introducing additional computational challenges. In this case, we are using ten assets to mitigate high computational time.

**Implementation and results.**

After establishing the parameters and assumptions, the portfolio optimization equation of the cost function was reformulated into a QUBO representation, facilitating its resolution through the QAOA. In this context, $X_i$ denotes the participation of asset i in the portfolio. The minimization equation encapsulates 10 variables along with a singular constraint, forming a structured problem definition.

In this instance, the cost function, articulated in the QUBO format, is disclosed to offer a snapshot into the massiveness and complexity of the equation, even when confined to merely 10 variables. This sight is intended to underscore the mathematical processes embedded in the computational process towards resolving an optimization problem. In the next cases, the remaining QUBO equations will not be showed for practical purposes.

**QUBO equation to select 5 from 10 assets:**

$$0.0001490\cdot x02+0.0001201\cdot x0x1+0.0001025\cdot x0x2+0.0001161\cdot x0x3+0.0000867\cdot x0x4+0.0001006\cdot x0x5+0.0000892\cdot x0x6$$
$$+0.0000861\cdot x0x7+0.0000890\cdot x0x8+0.0000902\cdot x0x9+0.0001121\cdot x12+0.0001345\cdot x1x2+0.0001553\cdot x1x3+0.0000932\cdot x1x4$$
$$+0.0001256\cdot x1x5+0.0001019\cdot x1x6+0.0001045\cdot x1x7+0.0001038\cdot x1x8+0.0001049\cdot x1x9+0.0001184\cdot x22+0.0001893\cdot x2x3$$
$$+0.0001180\cdot x2x4+0.0001607\cdot x2x5+0.0001474\cdot x2x6+0.0001295\cdot x2x7+0.0001096\cdot x2x8+0.0001219\cdot x2x9+0.0002052\cdot x32$$
$$+0.0001004\cdot x3x4+0.0001690\cdot x3x5+0.0001500\cdot x3x6+0.0001360\cdot x3x7+0.0001836\cdot x3x8+0.0001100\cdot x3x9+0.0001298\cdot x42$$
$$+0.0001164\cdot x4x5+0.0001439\cdot x4x6+0.0001130\cdot x4x7+0.0000663\cdot x4x8+0.0001077\cdot x4x9+0.0001155\cdot x52+0.0001394\cdot x5x6$$
$$+0.0001249\cdot x5x7+0.0000985\cdot x5x8+0.0001217\cdot x5x9+0.0002017\cdot x62+0.0001228\cdot x6x7+0.0000763\cdot x6x8+0.0001061\cdot x6x9$$
$$+0.0001429\cdot x72+0.0000937\cdot x7x8+0.0001334\cdot x7x9+0.0002109\cdot x82+0.0000711\cdot x8x9+0.0001021\cdot x92-0.0008146\cdot x0$$
$$-0.0006548\cdot x1-0.0007141\cdot x2-0.0009727\cdot x3-0.0005795\cdot x4-0.0007444\cdot x5-0.0006401\cdot x6-0.0005394\cdot x7-0.0008744\cdot x8$$
$$-0.0001587\cdot x9$$

Consequent to the computational simulations executed under the prescribed parameters, the resultant optimized portfolio has been determined to summarize the following stocks:

"ABBV", "ABT", "ADBE", "ADP", and "ATVI". This consolidation of assets manifests as the most favorable scenario, carefully balancing average returns and variance, associated with the derived data and predetermined constraints conditions. Moreover, it accurately adheres to the constraint of allocating investments into an exact subset of five, selected from the ten assets within the analyzed data set.

*Table 11 Feasible solutions of QAOA to select 5 from 10 assets with high correlation, and level of energy and probability.*

| Solution | Binary vector | Fval | Probability |
|---|---|---|---|
| Optimal | [1, 0, 0, 1, 1, 1, 0, 0, 1, 0 ] | -0,002 | 0,0019 |
| Feasible but non optimal solution | [1. 1. 0. 0. 0. 1. 1. 0. 1. 0.] | -0,00189 | 0,00098 |
| Feasible but non optimal solution | [1. 0. 0. 1. 0. 1. 0. 1. 1. 0.] | -0,00192 | 0,00195 |
| Feasible but non optimal solution | [1. 1. 1. 1. 0. 0. 0. 0. 1. 0.] | -0,00193 | 0,00195 |
| Feasible but non optimal solution | [1. 1. 0. 0. 1. 1. 0. 0. 1. 0.] | -0,00195 | 0,00293 |
| Feasible but non optimal solution | [1. 1. 1. 0. 0. 1. 0. 0. 1. 0.] | -0,00195 | 0,00195 |
| Feasible but non optimal solution | [1. 0. 0. 1. 0. 1. 1. 0. 1. 0.] | -0,00195 | 0,00293 |
| Feasible but non optimal solution | [1. 0. 1. 0. 1. 1. 0. 0. 1. 0.] | -0,00196 | 0,00098 |
| Feasible but non optimal solution | [1. 1. 0. 1. 1. 0. 0. 0. 1. 0.] | -0,00197 | 0,00195 |
| Feasible but non optimal solution | [1. 0. 1. 1. 1. 0. 0. 0. 1. 0.] | -0,00198 | 0,00391 |

*Table 12 No feasible solutions from QAOA to select 5 from 10 assets, and level of energy with probability.*

| Solution | Binary vector | Fval | Probability |
|---|---|---|---|
| No feasible solution | [0. 0. 0. 1. 0. 1. 0. 0. 1. 0.] | -0,00161 | 0,00195 |
| No feasible solution | [1. 0. 1. 1. 0. 0. 0. 0. 0. 0.] | -0,00162 | 0,00195 |
| No feasible solution | [1. 1. 1. 1. 0. 1. 0. 1. 1. 0.] | -0,00163 | 0,00098 |
| No feasible solution | [1. 1. 1. 1. 0. 1. 1. 0. 1. 0.] | -0,00164 | 0,00195 |

| | | | |
|---|---|---|---|
| No feasible solution | [1. 1. 1. 0. 1. 1. 0. 1. 1. 0.] | -0,00167 | 0,00098 |
| No feasible solution | [1. 0. 0. 0. 0. 1. 0. 0. 1. 0.] | -0,00167 | 0,00098 |
| No feasible solution | [1. 0. 1. 1. 1. 1. 0. 1. 1. 0.] | -0,00168 | 0,00098 |
| No feasible solution | [1. 0. 0. 1. 0. 0. 0. 0. 1. 0.] | -0,00171 | 0,00098 |
| No feasible solution | [1. 1. 1. 1. 1. 1. 0. 0. 1. 0.] | -0,00177 | 0,00195 |
| No feasible solution | [1. 0. 1. 1. 1. 0. 0. 1. 1. 0.] | -0,00182 | 0,00098 |

**Interpretation of Optimal Selection:**

Selection vector: The optimal solution vector [1, 0, 0, 1, 1, 1, 0, 0, 1, 0 ], serves as an indicative tool for asset inclusion within the optimal portfolio. Specifically, a value of 1 at a given position suggests the inclusion of the corresponding asset in the optimal mix. In the given context, assets situated at the first, fourth, fifth, sixth, and ninth positions from the dataset are earmarked for selection, aligning with the optimal portfolio construction.

**Quantum outcome analysis**: In the deployment of the Quantum Approximate Optimization Algorithm (QAOA), the search fundamentally aligns with identifying a minimum energy state of the Hamiltonian that satisfies the designated problem constraints, which, in the context of portfolio optimization, includes risk and return equilibriums. Upon combination with alternative potential or other feasible solutions, the derived optimal portfolio vector, with a probability of 0.0019, might suggest an infrequent occurrence within the quantum simulation. However, this specific outcome, though not the most recurrent, elucidates an optimal trade-off between risk and return, in adherence to the prescribed objective function.

**Probabilistic interpretations:** Engaging with quantum simulations it is important to mention the difference between occurrence frequency and optimality, especially within the framework of QAOA, where the algorithm try to find the lowest energy state in adherence to the imposed constraints. While certain portfolio configurations might materialize with heightened probabilities within the quantum simulation, it is essential to recognize that their prevalence does not inherently connote their optimality in aligning with the objective function. Consequently, even though certain outcomes might be feasible and recurrent, they might not invariably represent the optimal solution int he risk-return trade-off, given the constraint considerations inherent to the optimization problem.

**How to build a portfolio:**

a. Asset Identification:

Detailing the selected assets: It is crucial to ascertain which assets correspond to positions 1, 4, 5, 6, and 9 in the dataset or asset list. This involves mapping the binary vector's "1" positions to the respective assets in the dataset, ensuring a clear understanding of which specific investments are being included in the optimal portfolio.

b. Capital Allocation:

- Equitable distribution approach: Given the presumption of equitable capital distribution among the selected assets, the total capital intended for investment should be uniformly divided among them.
- Illustrative allocation: For instance, should the total capital be $10,000, an allocation of $2,000 per selected asset would be implemented, ensuring that each investment is given an equal financial weighting within the portfolio.

c. Asset Purchase:

- Execution of investment: Subsequent to the capital allocation, the next step involves the acquisition of shares or units of the identified assets, adhering closely to the allocated capital amounts.
- Strategic investment: Ensuring that the purchase adheres to the stipulated allocation facilitates alignment with the optimal portfolio construction derived from the model.

d. Rebalancing:

- Periodic portfolio review: Similar to methodologies employed in the Markowitz framework, the portfolio necessitates regular reviews to ensure its alignment with the investor's objectives and risk tolerance.
- Ensuring continued optimality: Rebalancing acts to ensure that the portfolio maintains its optimality in the evolving market conditions, asset performances, and any alterations in the investor's financial landscape or investment objectives.

**5.4 Simulation 3**

In this scenario, we utilized additional stock to address the optimization problem, the 20 first assets from SP500, employing the following parameters to execute the QAOA on a desktop computer and Google Colab.

*Table 13 Parameters for QAOA simulation 3.*

| Parameter | Value |
|---|---|
| Number of Assets (n) | 20 |
| Risk Factor (q) | 0.5 |
| Budget: (Number of Assets/2) | 10 |
| Penalty: Equal to the number of assets, used to scale the budget penalty term. | 20 |

Shortly after the initiation of the optimization process in Googel Colab and a desktop computer, several errors emerged:

- "ValueError: 'to_matrix' will return an exponentially large matrix, in this case '1048576x1048576' elements."
- "MemoryError: Unable to allocate 8.00 PiB for an array with shape (1125899906842625,) and data type uint64"

The error messages suggest that this simulation encountered a memory limitation while attempting to allocate a large array. Specifically, the array required 8 Pebibytes (PiB) of memory, which exceeds the capabilities of typical machines.

Potential solutions involved modifying the QAOA parameters, including reducing the number of iterations from 250 to 1, and adjusting the depth of the quantum circuit (reps) from 3 to 1. While decreasing the maximum iterations and reps can expedite computation, it might compromise the quality of the optimization. Even with these adjustments and an upgrade to Google Colab Pro, running the algorithm was not possible.

Ultimately, the code was executed directly on the Quantum Lab from IBM, where we were able to run it with 500 iterations and 3 reps to obtain the subsequent results for our optimization problem:

The QAOA solution yielded a binary outcome, wherein the positions of ones correspond to the assets: "ABBV", "ABT", "ADBE", "ADM", "AES", "ALB", "ALE", "AOS", "APD", "MMM". It's vital to note that this algorithm searches and highlights for the best approximation to the optimization problem and shows approximations that meet (feasible) and do not meet (infeasible) the constraints. Nonetheless, observing both feasible and infeasible solutions is crucial for comprehending the entire landscape of possibilities and make future decisions regarding to the strategy to select a portfolio.

*Table 14 Feasible solutions of QAOA simulation 3 to select 10 from 20 assets with high correlation, and level of energy Fvalue and probability.*

| Type of solution | Binary vector | Fvalue | Probability |
|---|---|---|---|
| Optimal | [0, 1, 0, 1, 0, 1, 1, 0, 1, 0,0, 1, 0, 0, 1, 1, 1, 0, 0, 1] | -0,006 | 0,002 |
| Feasible but non optimal solution | [1, 0, 1, 0, 1, 0, 0, 1, 0, 1,1, 0, 1, 1, 0, 0, 0, 1, 1, 0] | 0,0023 | 0,001 |
| Feasible but non optimal solution | [1, 0, 0, 0, 0, 1, 1, 1, 1, 0,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,0027 | 0,001 |
| Feasible but non optimal solution | [0, 0, 0, 1, 1, 1, 1, 0, 0, 0,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,0027 | 0,001 |
| Feasible but non optimal solution | [0, 0, 0, 1, 1, 0, 1, 0, 0, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00378 | 0,001 |
| Feasible but non optimal solution | [1, 0, 0, 0, 0, 0, 1, 1, 1, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,0047 | 0,001 |
| Feasible but non optimal solution | [0, 1, 0, 1, 0, 1, 1, 0, 0, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,0025 | 0,001 |
| Feasible but non optimal solution | [0, 1, 1, 1, 1, 0, 1, 0, 0, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,0027 | 0,001 |
| Feasible but non optimal solution | [1, 0, 1, 0, 0, 0, 1, 0, 1, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,0094 | 0,001 |
| Feasible but non optimal solution | [1, 0, 1, 0, 1, 0, 1, 1, 1, 0,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,0043 | 0,001 |

**Section 2**

**5. 5 Simulation 4 and 5 with assets that have low correlation.**

In the previous cases for simulations 1.1, 1.2, 2 and 3 , the dataset comprised the first 10 and 20 assets from the SP&500; even though they are from different market sectors, we noticed something important: the assets were highly correlated. This implies that they tend to move together in the same direction, which is not advisable for an investor trying to diversify their portfolio to manage risk effectively. Diversification involves trying to select a mix of assets that will not all rise or fall at the same time, thereby reducing the risk of significant losses without sacrificing too much in potential gains. Having stocks with high correlation in one portfolio does not yield the benefits of diversification because when one asset loses value, others may follow that trend.

In the new scenario, 10 assets from the S&P 500 that have the lowest correlation with each other were selected. Essentially, we choose assets that are less likely to move up or down together. We are then applying three different methods - Markowitz's and Black-Litterman model, and QAOA - to provide us with different perspectives on our portfolio optimization.

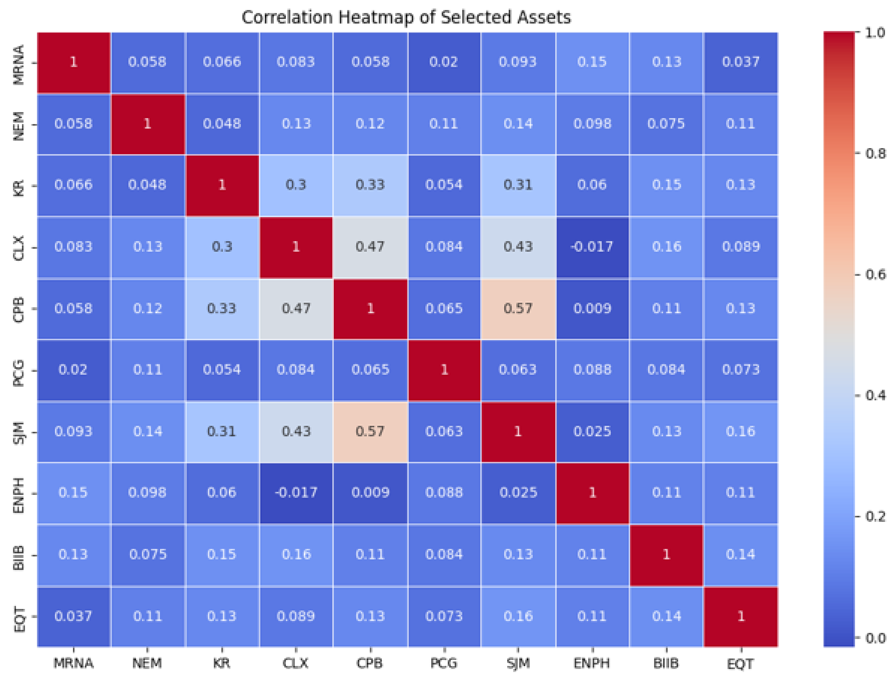


*Figure 9 Correlation matrix of stocks with low correlation.*

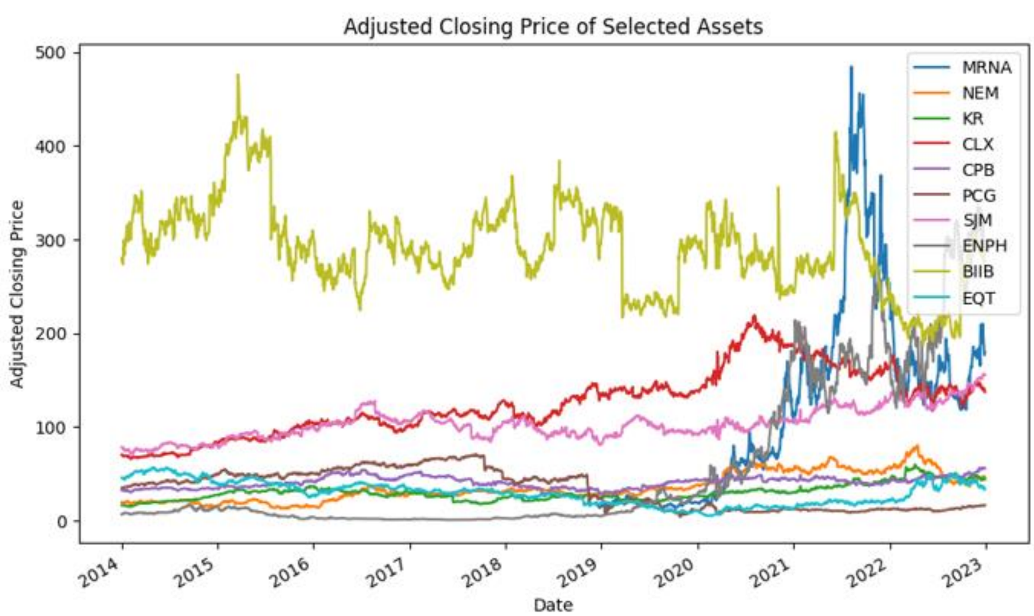


*Figure 10 Adjusted closing price of stocks with low correlation.*

### 5.6 Simulation 4

#### 5.6.1 Markowitz and Black-Litterman simulation 4.1

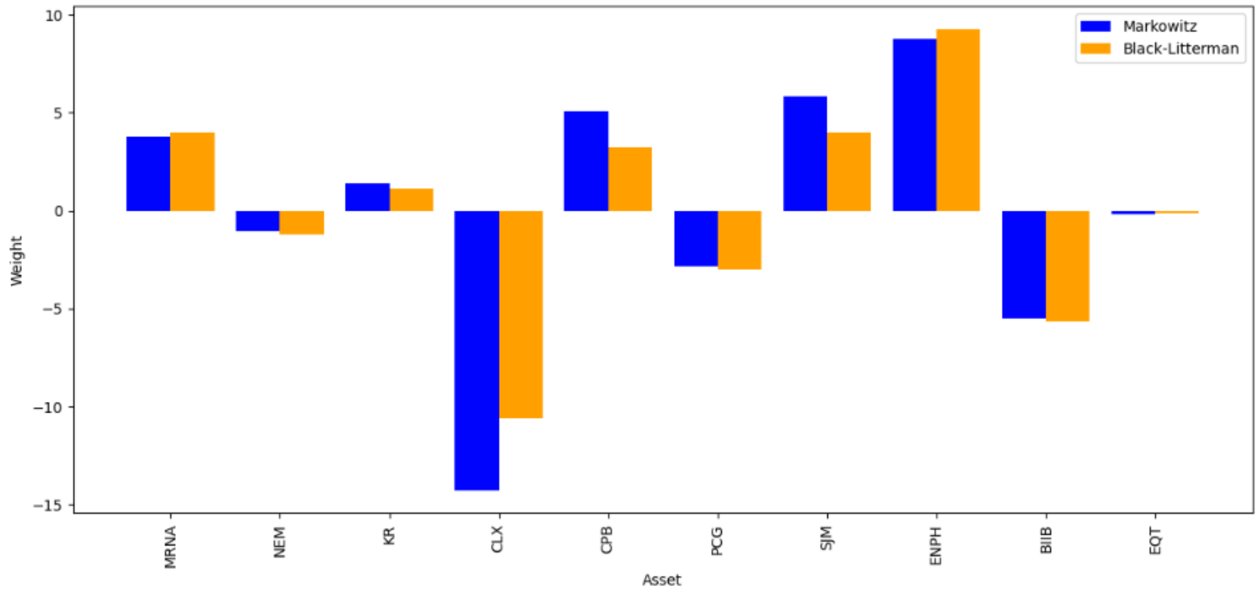


*Figure 11 Optimal portfolio weights: Markowitz vs Black-Litterman Simulation 4.1.*

*Table 15 Markowitz and Black-Litterman weights Simulation 4.1*

| Asset | Wi Markowtiz | Wi Black-Litterman |
|---|---|---|
| MRNA | 3,78 | 3,97 |
| NEM | -1,06 | -1,20 |
| KR | 1,41 | 1,13 |
| CLX | -14,25 | -10,56 |
| CPB | 5,05 | 3,21 |
| PCG | -2,82 | -3,00 |
| SJM | 5,84 | 3,98 |
| ENPH | 8,75 | 9,24 |
| BIIB | -5,51 | -5,65 |
| EQT | -0,19 | -0,12 |

### 5.6.2 Markowitz and Black Litterman simulation 4.2

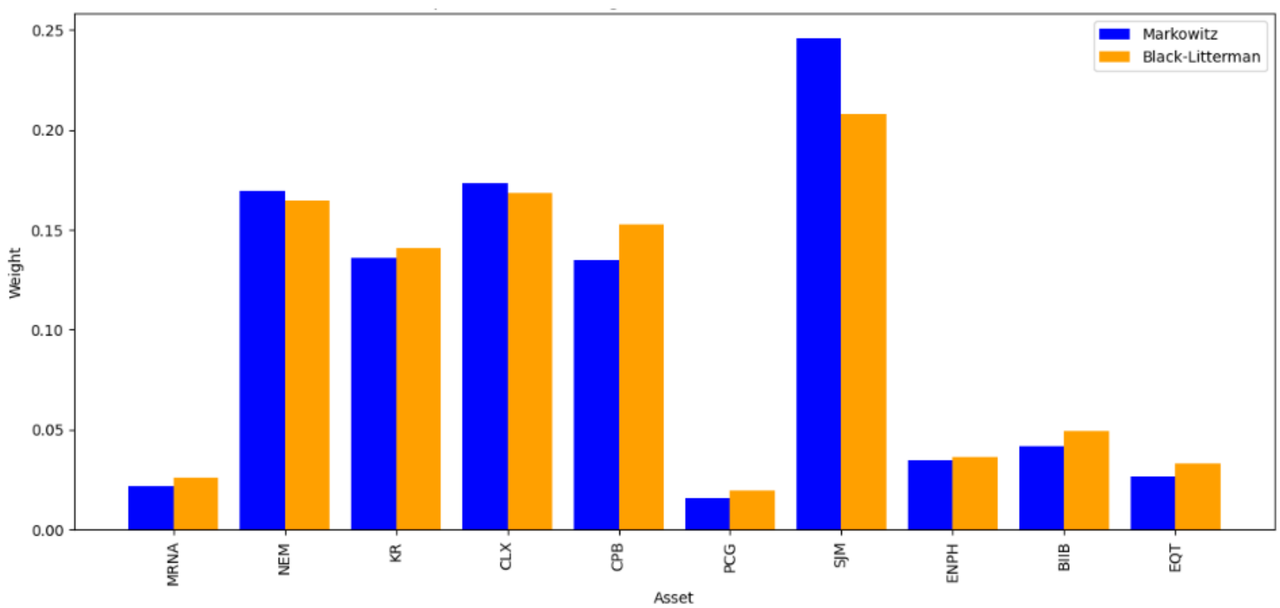


*Figure 12 Optimal portfolio weights: Markowitz vs Black-Litterman Simulation 4.2.*

*Table 16 Markowitz and Black-Litterman weights Simulation 4.2.*

| Asset | Wi Markowitz | Wi Black-Litterman |
|---|---|---|
| MRNA | 0,022 | 0,026 |
| NEM | 0,169 | 0,164 |
| KR | 0,136 | 0,141 |
| CLX | 0,173 | 0,169 |
| CPB | 0,135 | 0,153 |
| PCG | 0,016 | 0,020 |
| SJM | 0,246 | 0,208 |
| ENPH | 0,035 | 0,037 |
| BIIB | 0,042 | 0,050 |
| EQT | 0,027 | 0,033 |

In this scenario, where assets with low correlations are employed, the Markowitz and Black-Litterman methods produce distinct weightings for portfolio selection. In this case, the views or opinions integrated into the Black-Litterman model yield different outcomes compared to Markowitz's approach. These divergent results can be attributed to the specific assumptions and methodologies inherent to each method, demonstrating the importance of understanding the underlying principles and implications when selecting a portfolio optimization strategy.

### 5.6.3 Simulation 5 QAOA to select 5 out of 10 assets.

*Table 17 Feasible solutions QAOA simulation 5 with assets with low correlation, and level of energy Fvalue and probability.*

| Type of solution | Binary vector | Fvalue | Probability |
|---|---|---|---|
| Optimal | [1. 1. 1. 0. 0. 0. 1. 1. 0. 0.] | -0,0065 | 0,0039 |
| Feasible but non optimal solution | [1. 1. 0. 0. 1. 0. 0. 1. 0. 1.] | -0,006 | 0,0039 |
| Feasible but non optimal solution | [1. 1. 0. 0. 0. 0. 1. 1. 0. 1.] | -0,0061 | 0,002 |
| Feasible but non optimal solution | [1. 0. 0. 0. 1. 0. 1. 1. 0. 0.] | -0,0064 | 0,001 |
| Feasible but non optimal solution | [1. 0. 1. 0. 1. 0. 0. 1. 0. 1.] | -0,0064 | 0,002 |
| Feasible but non optimal solution | [1. 0. 1. 0. 0. 0. 1. 1. 0. 1.] | -0,0064 | 0,002 |
| Feasible but non optimal solution | [1. 1. 1. 0. 1. 0. 0. 1. 0. 0.] | -0,0065 | 0,0029 |
| Feasible but non optimal solution | [1. 1. 1. 0. 1. 0. 0. 1. 0. 1.] | -0,0065 | 0,0039 |
| Feasible but non optimal solution | [1. 0. 0. 0. 1. 0. 1. 1. 0. 1.] | -0,0065 | 0,0029 |
| Feasible but non optimal solution | [1. 1. 0. 0. 1. 0. 1. 1. 0. 0.] | -0,0065 | 0,0029 |

The optimal solution that satisfies the constraints in this case are MRNA, NEM, KR, SJM, and ENPH to select 5 of ten assets to build the optimal portfolio.

Assets 1 and 8 consistently feature in all 10 solutions, suggesting their substantial presence in both optimal and suboptimal portfolios. This insight offers a foundational understanding of the potential significance of these assets within the portfolio across diverse scenarios. Assets with frequent appearances across various solutions may play a pivotal role in achieving our optimization objectives.

## 6. Comparative analysis and backtesting strategy.

a. Initial Investment:

To this analysis a theoretical initial investment of $10,000 is selected, either distributed across various assets of each method. This predefined financial allocation serves as a constant, enabling a standardized assessment of various weights allocations for Markowitz and Black-Litterman optimization, and asset allocation strategies in QAOA methodology, to help in measuring their respective impacts on portfolio performance.

b. Performance Metrics:

- Cumulative Returns: This metric summarizes the comprehensive return achieved by the portfolio over a defined temporal period, serving as a quantitative representation of the portfolio's financial performance.
- Sharpe Ratio: The Sharpe Ratio quantifies the risk-adjusted returns of the portfolio, providing a normalized measure of return relative to risk, thereby facilitating an objective evaluation of risk-management efficacy.

c. Comparison with Benchmark ( S&P 500):

The analytical framework requires a comparison with a predefined benchmark, in this case, the S&P 500. This comparative analysis enables the assessment of the portfolio's performance relative to a widely recognized market standard, providing a relative measure of effectiveness and performance.

d. Visualizing the results:

- Visualization of both cumulative returns and portfolio values over time provides a visual comparative analysis against the S&P 500, enabling a graphical interpretation of relative performance trajectories.
- Additional visual analyses involving risk and return trade-offs serves to graphically represent the portfolio’s financial resilience and stability of the optimal portfolio vs non optimal portfolios.
- It's important to note that when using QAOA for asset allocation, all assets are assigned equal weightage, in contrast to methods like Markowitz and Black-Litterman, which allow for individual asset weight determination. In the context of this study, the QAOA method is employed exclusively for identifying the optimal asset selection, which is why all assets are assigned equal weightage.
- When backtesting the results of QAOA, Portfolio 1 is identified as the optimal solution, while Portfolios 2 to 9 are considered as portfolios with feasible solutions but not optimal ones.

**6. 1. Backtesting the results of the Markowitz portfolio optimization, simulation 1.2**

In this scenario, backtesting analysis is restricted to the Markowitz Model 1.2 because the results from the Black-Litterman Model displayed similarity to the weights derived from Markowitz's approach. Additionally, Simulation 1.1 is not subjected to backtesting, as this study does not explore negative weights or short selling. This is because this portfolio optimization study is intended to be applied to long-term investments.

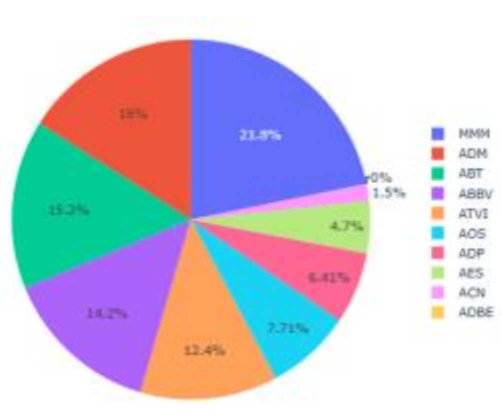


*Figure 13 Portfolio weight allocation simulation 1.2 using Markowitz optimization.*


*Figure 14 Portfolio cumulative return simulation 1.2 Markowitz optimization.*

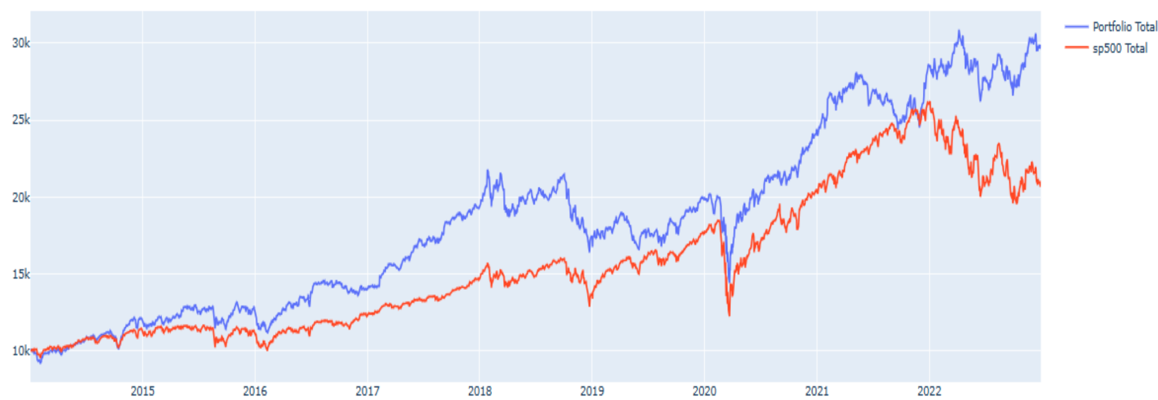


*Figure 15 Portfolio value vs SP500. Simulation 1.2, Markowitz optimization.*

### 6.2. Backtesting simulation 2 QAOA to select 5 optimal assets from 10 highly correlated stocks.

*Table 48 QAOA simulation 2. feasible solutions for selecting 5 assets from a set of 10.*

| Type of solution | Portfolio | Binary vector | Expected Return | Volatility |
|---|---|---|---|---|
| Optimal | 1 | [1, 0, 0, 1, 1, 1, 0, 0, 1, 0 ] | 0,00317 | 0,06241 |
| Feasible but non optimal solution | 2 | [1. 1. 0. 0. 0. 1. 1. 0. 1. 0.] | 0,00294 | 0,06082 |
| Feasible but non optimal solution | 3 | [1. 0. 0. 1. 0. 1. 0. 1. 1. 0.] | 0,00312 | 0,06373 |
| Feasible but non optimal solution | 4 | [1. 1. 1. 1. 0. 0. 0. 0. 1. 0.] | 0,00323 | 0,06489 |
| Feasible but non optimal solution | 5 | [1. 1. 0. 0. 1. 1. 0. 0. 1. 0.] | 0,00295 | 0,05873 |

| | | | | |
|---|---|---|---|---|
| Feasible but non optimal solution | 6 | [1. 1. 1. 0. 0. 1. 0. 0. 1. 0.] | 0,00310 | 0,06093 |
| Feasible but non optimal solution | 7 | [1. 0. 0. 1. 0. 1. 1. 0. 1. 0.] | 0,00316 | 0,06503 |
| Feasible but non optimal solution | 8 | [1. 0. 1. 0. 1. 1. 0. 0. 1. 0.] | 0,00300 | 0,05964 |
| Feasible but non optimal solution | 9 | [1. 1. 0. 1. 1. 0. 0. 0. 1. 0.] | 0,00309 | 0,06211 |
| Feasible but non optimal solution | 10 | [1. 0. 1. 1. 1. 0. 0. 0. 1. 0.] | 0,00314 | 0,06296 |

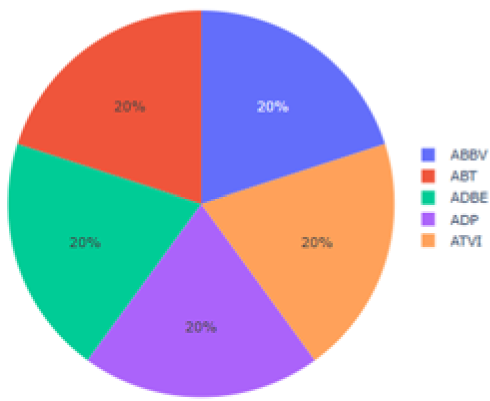


*Figure 16 Portfolio weight allocation simulation 2 using QAOA.*

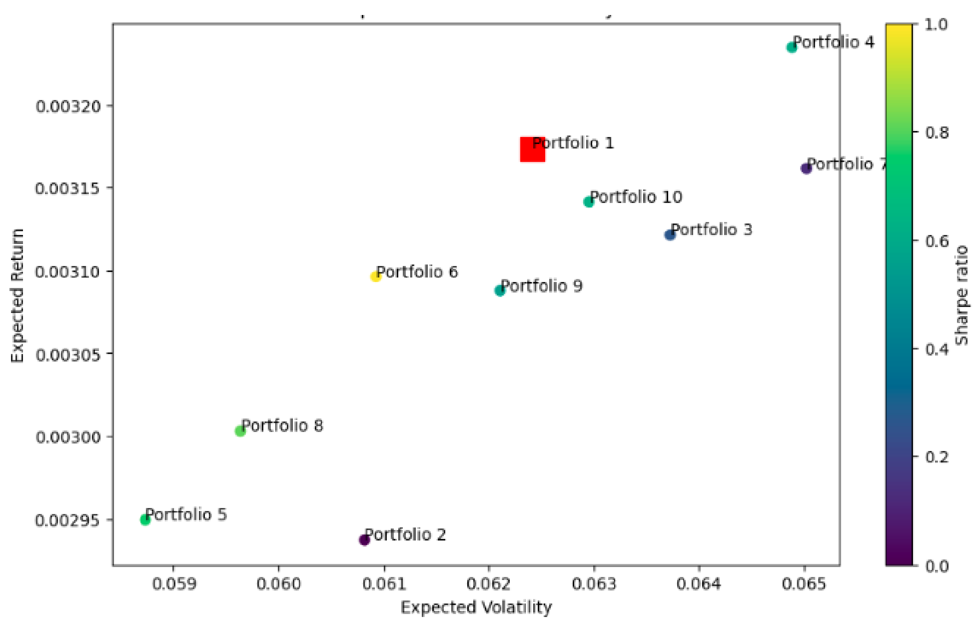


*Figure 17 Trade-off expected return vs expected volatility of QAOA results from simulation 2.*

Figure 17 presents a scatter plot, illustrating the points correlated with each feasible solution within the objective space, where the objectives are the expected return and volatility. This visual representation aids in pinpointing the efficient frontier, which encompasses the points related to solutions from portfolio 1 to 10. The optimal portfolio 1 is labeled as efficient since it is not surpassed by any other solutions. Considering the investigation conducted by Gomez (2022), a general solution A is considered to dominate solution B if A possesses a lower volatility at the same expected return as B, or if A has a heightened expected return at the same volatility as B, or even if A simultaneously exhibits lower volatility and superior expected return compared to B.

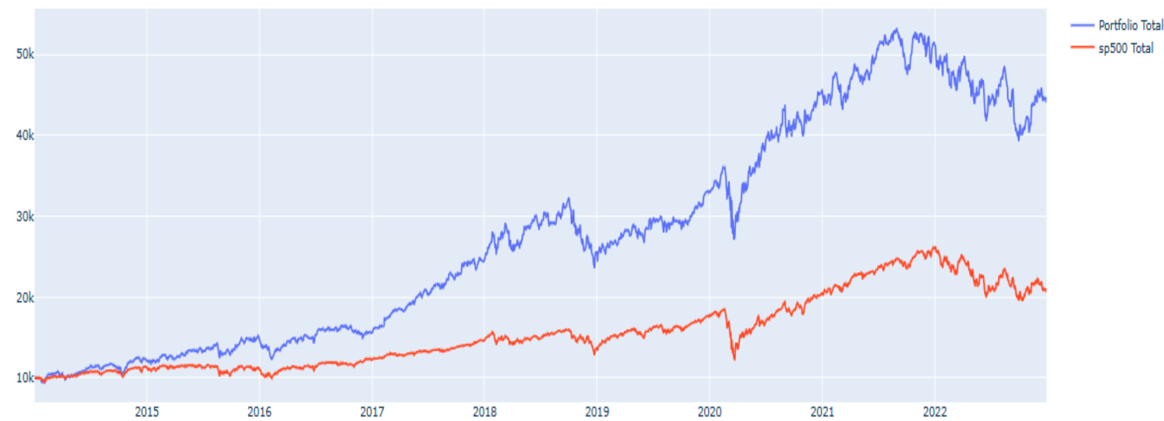


*Figure 18 Evolution of portfolio 1 value in simulation 2 QAOA.*

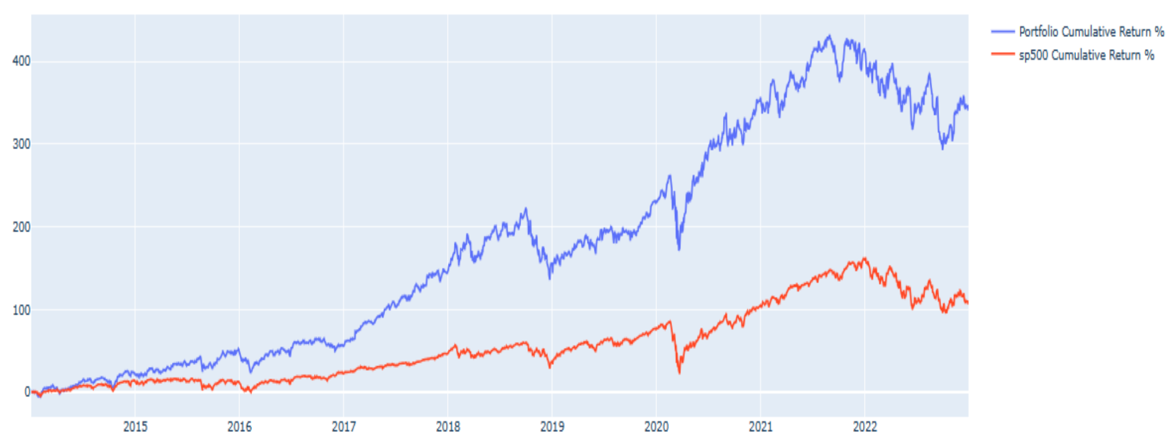


*Figure 19 Portfolio 1 cumulative return vs SP500. Simulation 2, QAOA.*

### 6.3 Backtesting simulation 3 QAOA for selecting 10 out of 20 stocks.


*Figure 20 Portfolio weight allocation simulation 3 using QAOA.*

*Table 19 Optimal and feasible solutions with Expected return and Volatility for simulation 3*

| Type of solution | Portfolio | Binary vector | Expected Return | Volatility |
|---|---|---|---|---|
| Optimal | 1 | [0, 1, 0, 1, 0, 1, 1, 0, 1, 0,0, 1, 0, 0, 1, 1, 1, 0, 0, 1] | 0,00492 | 0,12222 |
| Feasible but non optimal solution | 2 | [1, 0, 1, 0, 1, 0, 0, 1, 0, 1,1, 0, 1, 1, 0, 0, 0, 1, 1, 0] | 0,00337 | 0,13941 |
| Feasible but non optimal solution | 3 | [1, 0, 0, 0, 0, 1, 1, 1, 1, 0,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00465 | 0,13359 |
| Feasible but non optimal solution | 4 | [0, 0, 0, 1, 1, 1, 1, 0, 0, 0,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00411 | 0,11924 |
| Feasible but non optimal solution | 5 | [0, 0, 0, 1, 1, 0, 1, 0, 0, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00378 | 0,11776 |
| Feasible but non optimal solution | 6 | [1, 0, 0, 0, 0, 0, 1, 1, 1, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00432 | 0,13252 |
| Feasible but non optimal solution | 7 | [0, 1, 0, 1, 0, 1, 1, 0, 0, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00462 | 0,12811 |
| Feasible but non optimal solution | 8 | [0, 1, 1, 1, 1, 0, 1, 0, 0, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00343 | 0,14675 |
| Feasible but non optimal solution | 9 | [1, 0, 1, 0, 0, 0, 1, 0, 1, 1,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00268 | 0,14242 |
| Feasible but non optimal solution | 10 | [1, 0, 1, 0, 1, 0, 1, 1, 1, 0,0, 0, 1, 1, 1, 0, 1, 0, 0, 1] | 0,00346 | 0,15397 |

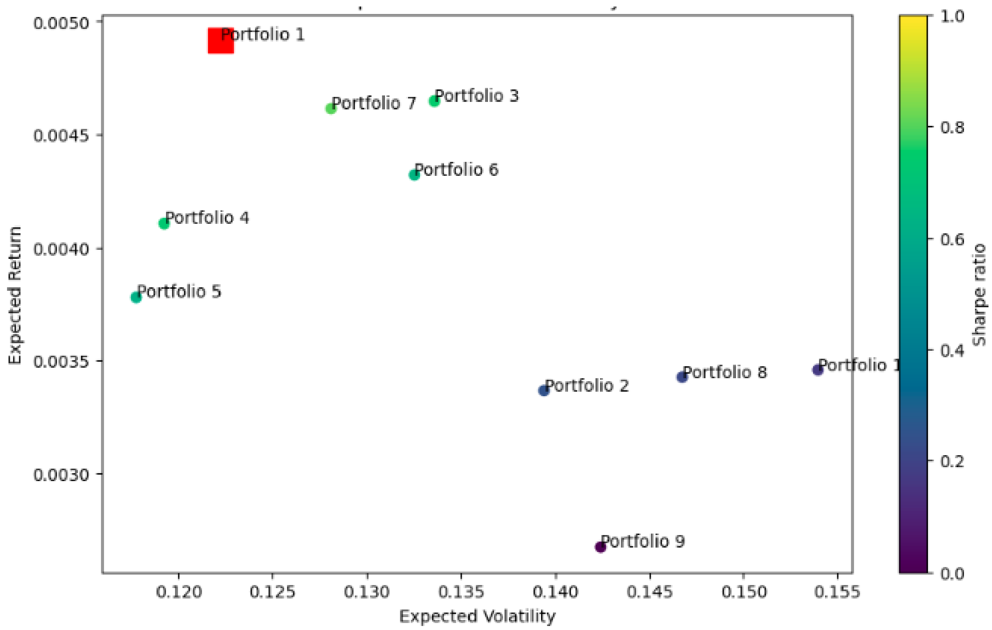


*Figure 21 Trade-off expected return vs expected volatility simulation 3.*

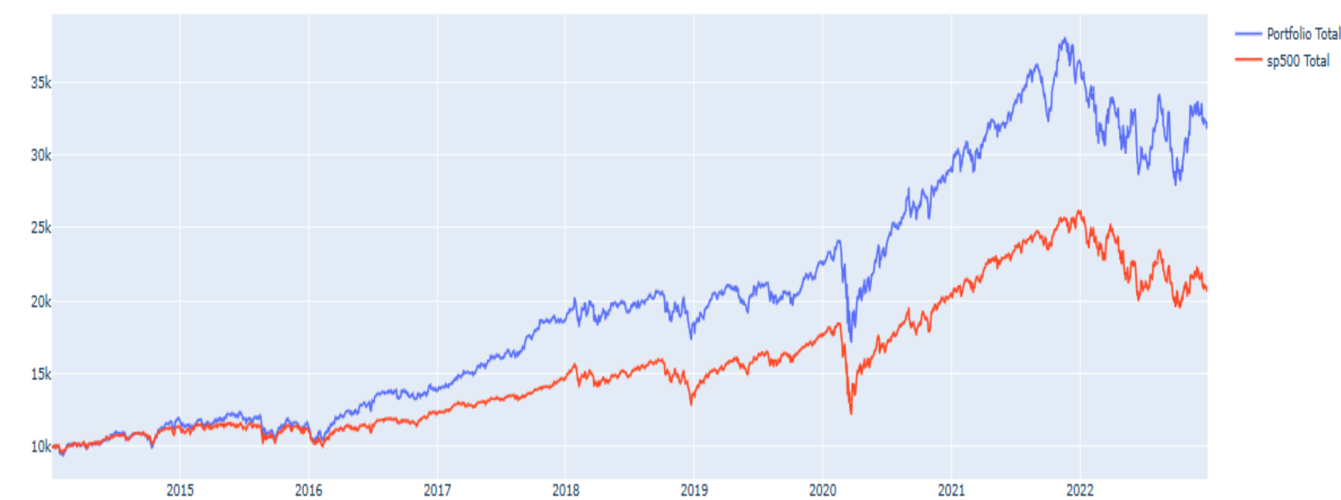


*Figure 22 Evolution of portfolio 1 value in simulation 3.*

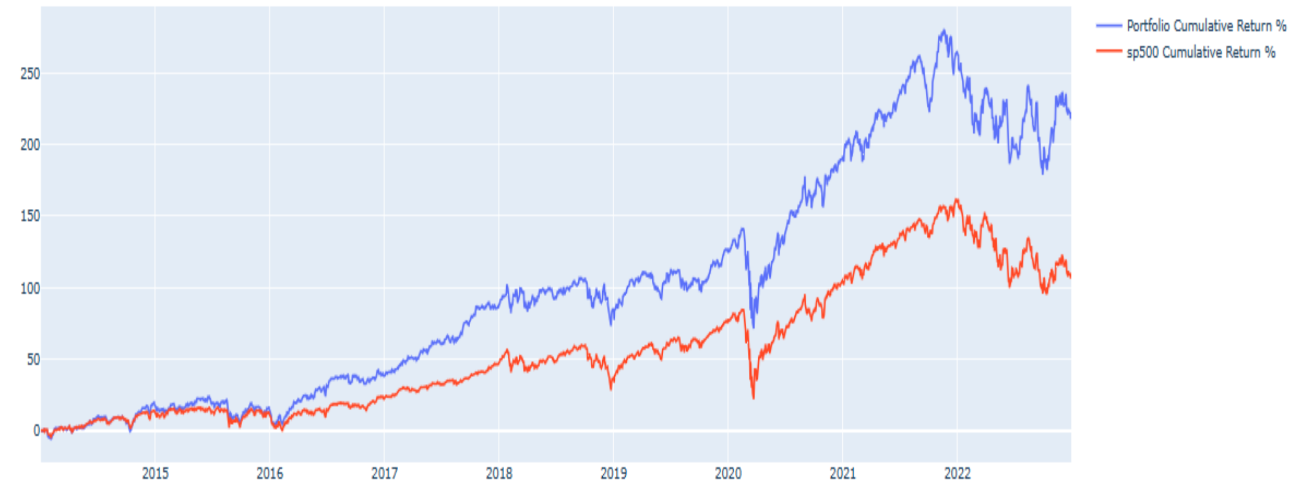


*Figure 23 Portfolio 1 cumulative return vs SP500 in simulation 3.*

In this case, the portfolio 1 selected as the optimal solution shows a better trade-off between expected return and volatility.

### 6.4 Scenario 2

#### 6.4.1 Backtesting results of the Markowitz and Black-Litterman simulation 4.

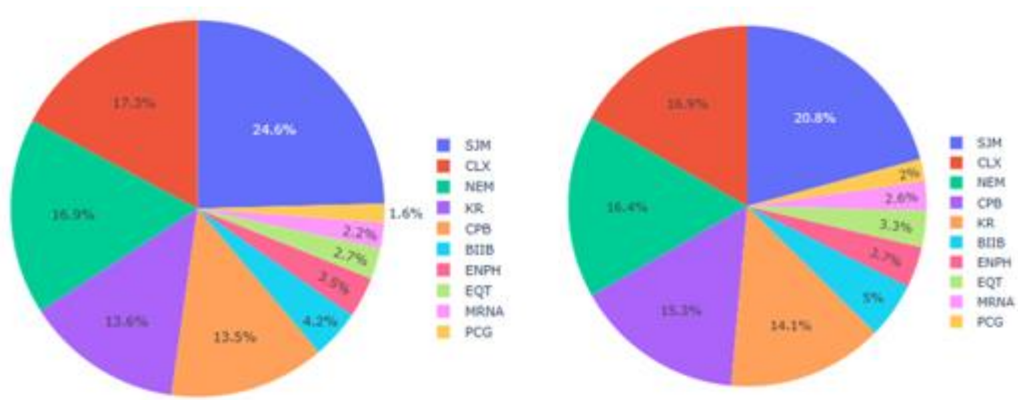


*Figure 24 Weights allocation with Markowitz (left) and Black-Litterman (right) optimization simulation 4.*

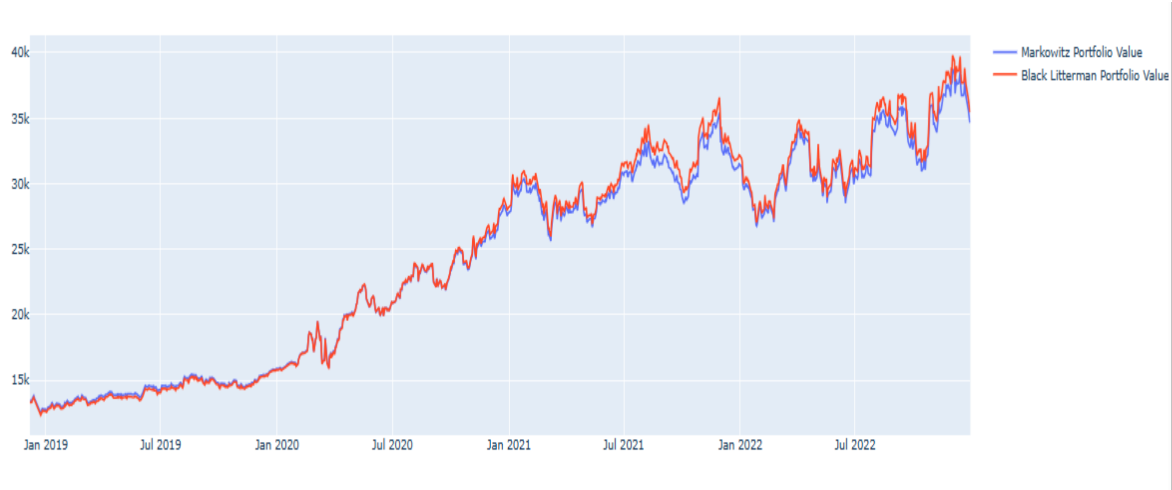


*Figure 25 Portfolio value over time Markowtiz (blue) vs Black-Litterman (red) simulation 4.*

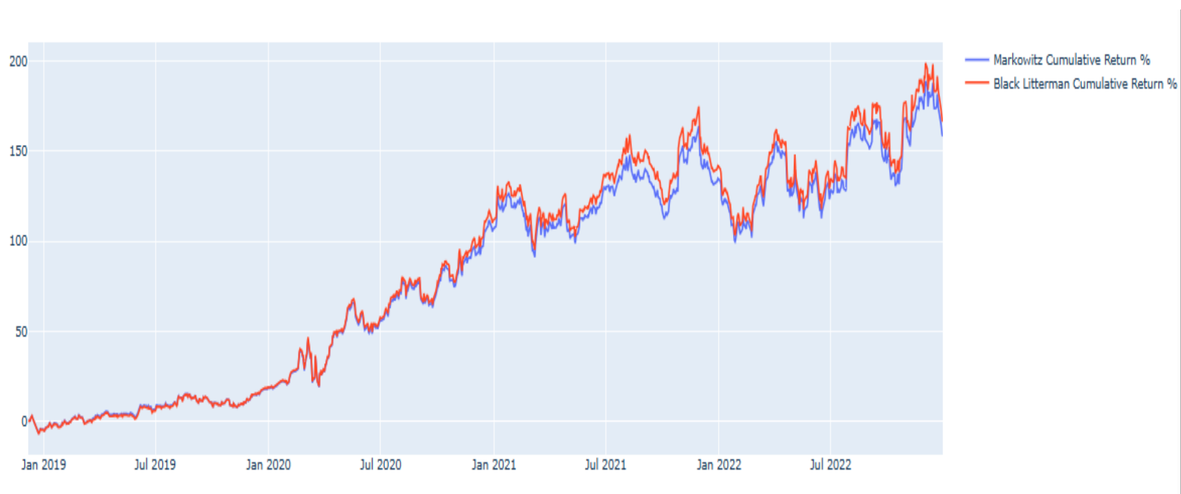


*Figure 26 Portfolio cumulative return over time Markowtiz (blue) vs Black-Litterman(red) simulation 4.*

In this scenario, the allocation of weights through the Black-Litterman model has proven to be more advantageous in terms of portfolio cumulative returns and overall portfolio value when compared to the traditional Markowitz method. This observation aligns with the principles of modern portfolio theory, where the incorporation of informed views within the Black-Litterman framework appears to outperform the outcomes generated solely by the Markowitz approach. This suggests that the utilization of the Black-Litterman model not only enhances portfolio performance but also underscores the importance of considering informed perspectives in the portfolio optimization process.

### 6.4.2 Backtesting simulation 5 QAOA to select 5 from 10 assets with low correlation.

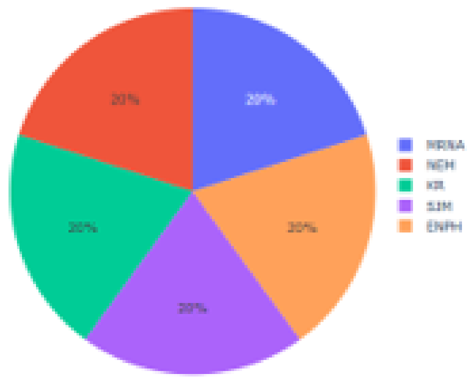


*Figure 27 Portfolio weight allocation simulation 5.*

*Table 20 Optimal and feasible solutions with expected return vs expected volatility in simulation 5.*

| Type of solution | Portfolio | Binary vector | Expected Return | Volatility |
|---|---|---|---|---|
| Optimal | 1 | [1. 1. 1. 0. 0. 0. 1. 1. 0. 0.] | 0,00312 | 0,07541 |
| Feasible but non optimal solution | 2 | [1. 1. 0. 0. 1. 0. 0. 1. 0. 1.] | 0,00083 | 0,06281 |
| Feasible but non optimal solution | 3 | [1. 1. 0. 0. 0. 0. 1. 1. 0. 1.] | 0,00319 | 0,07554 |
| Feasible but non optimal solution | 4 | [1. 0. 0. 0. 1. 0. 1. 1. 0. 0.] | 0,00245 | 0,07727 |
| Feasible but non optimal solution | 5 | [1. 0. 1. 0. 1. 0. 0. 1. 0. 1.] | 0,00077 | 0,06364 |
| Feasible but non optimal solution | 6 | [1. 0. 1. 0. 0. 0. 1. 1. 0. 1.] | 0,00313 | 0,07571 |
| Feasible but non optimal solution | 7 | [1. 1. 1. 0. 1. 0. 0. 1. 0. 0.] | 0,00077 | 0,06286 |
| Feasible but non optimal solution | 8 | [1. 1. 1. 0. 1. 0. 0. 1. 0. 1.] | 0,00107 | 0,07008 |
| Feasible but non optimal solution | 9 | [1. 0. 0. 0. 1. 0. 1. 1. 0. 1.] | 0,00275 | 0,08160 |
| Feasible but non optimal solution | 10 | [1. 1. 0. 0. 1. 0. 1. 1. 0. 0.] | 0,00275 | 0,08123 |

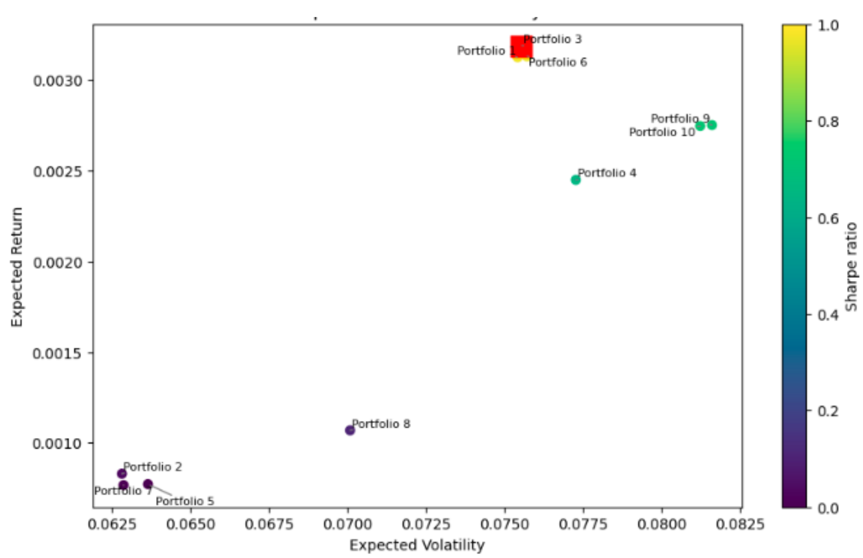


*Figure 28 Trade-off expected return vs expected volatility in simulation 5.*

The Quantum Approximate Optimization Algorithm (QAOA) indeed produces a spectrum of results, encompassing both feasible and non-feasible solutions to optimization problems. Among these feasible solutions, the algorithm identifies the optimal solution while also providing alternatives that adhere to the specified constraints. An example of such a constraint is the risk aversion parameter, denoted as q, which impacts the delicate balance between expected return and portfolio volatility. The possibility to access and evaluate all feasible solutions offers investors a comprehensive view of potential outcomes, facilitating the process of making informed decisions when constructing a portfolio. For instance, consider the scenario depicted in the figure 27, where two distinct portfolios, labeled as Portfolio 1 and Portfolio 3, closely align with the optimal solution while respecting the predefined constraints. In this context, Portfolio 1 was selected as the solution by the QAOA results that best aligns with these constraints, and Portfolio 3 exhibits slightly higher expected returns, although with a slightly elevated level of volatility. This information equips investors with the insights needed to carefully assess their options between these two portfolios. Moreover, it's crucial to recognize that the decision-making process is not solely influenced by the algorithm's outputs; it is also shaped by individual perspectives and expectations associated with the assets contained within Portfolios 1 and 3.

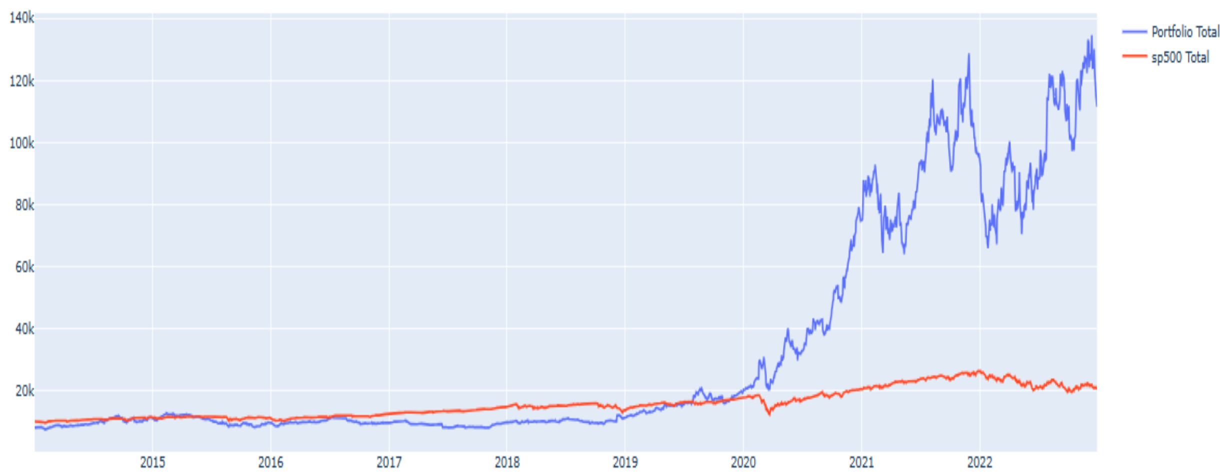


*Figure 29 Optimal portfolio 1 value vs SP500 in simulation 5*

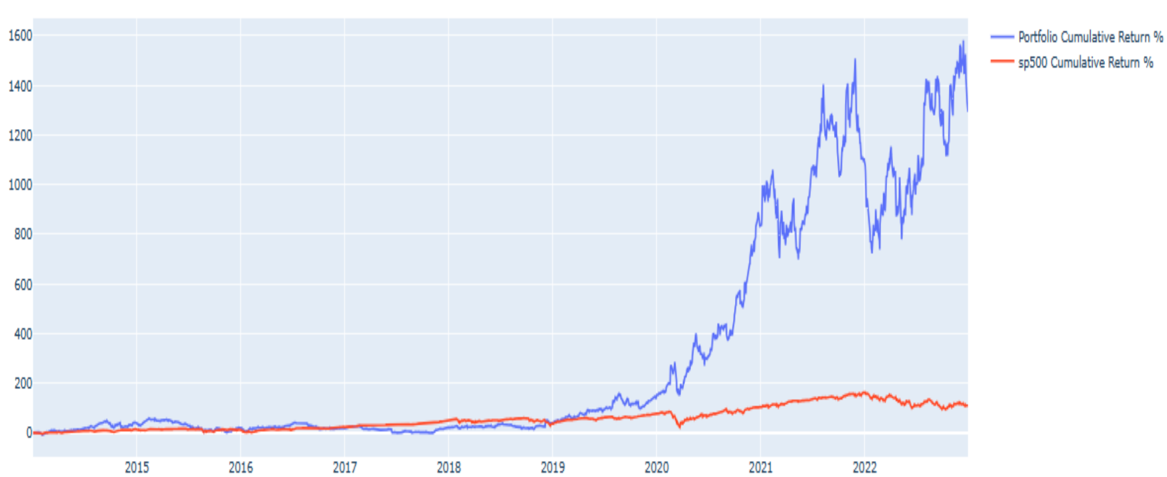


*Figure 30 Portfolio 1 cumulative return vs SP500 in simulation 5.*

## 7. Conclusions

The backtesting process provides valuable insights into the cumulative return percentage of the optimal portfolios chosen and the portfolio value over time, showing the effectiveness of different portfolio optimization strategies under specified constraints. Specifically, these strategies involve the utilization of Markowitz and Black-Litterman methods for weight allocation and the Quantum Approximate Optimization Algorithm (QAOA) for asset selection.

In conclusion, the application of three distinct portfolio optimization methodologies has yielded portfolios that consistently demonstrated a performance advantage over the S&P 500 benchmark in terms of cumulative returns. This consistent outperformance across all three methods signifies their effectiveness in identifying portfolios that generated superior returns when compared to the broader market represented by the S&P 500.

Furthermore, while cumulative returns provide a valuable measure of investment success, incorporating additional performance metrics like the Sharpe Ratio and visual analysis of portfolio trajectories provided a broader view of portfolio performance and help guide investment decision-making.

In summary, the consistent outperformance of portfolios selected through Markowitz, and Black-Litterman and QAOA in comparison to the S&P 500 benchmark underlines their ability to effectively identify and allocate assets for superior returns, demonstrating their value in the field of portfolio optimization and investment management.

The application of these strategies across multiple simulations highlights their empirical utility as decision-making tools for investors facing various scenarios. Here's a more detailed exploration of these key points:

**a. Optimization strategies**: The simulations encompass a comprehensive exploration of portfolio optimization techniques. Markowitz and Black-Litterman approaches are employed for the allocation of portfolio weights, while the QAOA is leveraged to select assets. These strategies collectively aim to achieve the optimal trade-off between return and risk mitigation.

**b. Cumulative return analysis**: The graph, which compares the portfolio's cumulative return percentage to that of the S&P 500, serves as a visual representation of the strategies' effectiveness. It allows investors to gauge how well their portfolios are performing relative to a widely recognized benchmark.

**c. Constraints**: The strategies employed in these simulations operate within predefined constraints. These constraints may include factors such as risk tolerance, asset allocation limits. Adhering to these constraints is crucial for aligning the portfolio with an investor's risk appetite and investment objectives.

**d. Data sets**: The data sets used in these simulations play an important role in the decision-making process. Accurate and reliable data is vital for making informed investment decisions. The strategies employed here are tested and validated using real-world data, enhancing their practical applicability.

**e. Scenario-based decision making**: The results generated from these simulations empower investors with a hybrid and empirical toolset for making decisions in diverse scenarios and meet the constraints of the investor profile.

it's important to acknowledge the limitations of the Quantum Approximate Optimization Algorithm (QAOA) within the context of portfolio optimization. While QAOA is a powerful hybrid approach that provides valuable insights and approximations to the optimal solution, it does come with its own set of limitations that investors should consider when selecting an appropriate strategy based on their profile and constraints:

**f. Approximate Solutions**: QAOA is designed to yield approximations to the optimal solution. While these approximations can be highly valuable, investors should be aware that increasing the number of assets increases the computational power needed to solve the optimization problem.

**g. Complexity**: QAOA's effectiveness depends on various factors, including the choice of quantum hardware, algorithmic parameters, and the specific problem instance. In real quantum programming environment tuning parameters of the quantum circuit and obtaining accurate results can be complex, potentially requiring specialized expertise or access to quantum computing resources.

**h. Computational resources**: Quantum computing capabilities may be limited, especially in scenarios of more than 10 assets retail investors or smaller asset management firms need to consider the access to quantum hardware and the computational power required for QAOA may pose a practical constraint for some investors.

## 8. Prospects for Future Research:

Recent studies have indicated that the performance of the standard QAOA can be enhanced through different techniques. For instance, the study developed by He (2022) mentions the local Hamiltonian-based QAOA and Dehn (2023) the application of warm-start approaches as a well-established method for reducing the computational time required to solve optimization problems, as it initiates optimization with an efficiently computable approximate solution. Hence, it would be beneficial for forthcoming investigations to explore further into the application of warm-start techniques within the context of QAOA, exploring their potential to expedite the optimization process and improve algorithmic efficiency. Other direction, for futures works, may be changing quantum processing unit parameters in the quantum algorithm to find better approximations to the solution of the optimization problem.

## 9. References


Moein Owhadi-Kareshk, Pierre Boulanger.(2021). "Portfolio Optimization on Classical and QuantumComputers Using PortFawn", Department of Computing Science.1-9.

Samuel Palmer,1 Serkan Sahin.(2021). "Quantum Portfolio Optimization with Investment Bands and Target Volatility", Multiverse Computing, Centre for Social Innovation.1-5.

N. Slate, E. Matwiejew, S. Marsh, and J. B. Wang.(2021) "Quantum walk-based portfolio optimization", Department of Physics, The University of Western Australia, Perth.1-17.

Gunjan, A and Bhattacharyya, S.(2022) "A brief review of portfolio optimization techniques", Artificial Intelligence Review, 1-5.

Gomez, A; Leitao, A; (...); Vazquez, C. (2022). "A Survey on Quantum Computational Finance for Derivatives Pricing and VaR". Archives of Computational Methods in Engineering.1-4.

Lee, RST. (2020)."AI Powerful Tools in Quantum Finance", Quantum Finance: Intelligent Forecast and Trading Systems.41-43, 159-162.

Sotelo, R .(2021) "Applications of quantum computing to optimization", IEEE Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (IEEE chilecon 2021).1-5.

Bova, F; Goldfarb, A and Melko, RG. (2021).Commercial Applications of Quantum Computing, EPJ Quantum Technology.

Ji, W and Meng, XD.(2020)." Demonstration of quantum linear equation solver on the IBM qiskit platform "19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES).1-4.

Mugel, S; Kuchkovsky, C; (...); Orus, R.(2022).“Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks”, Physical Review Research.1-5.

Nastasiuk, VA. (2014).“Emergent quantum mechanics of finances”, Physic a-statistical Mechanics and its Applications.1-7.

Rebentrost, P; Hamoudi, Y; (...); Santha, M .(2021)“Quantum algorithms for hedging and the learning of Ising models”, Physical Review A.1-5.

Baaquie, BE.(2018) “Quantum Field Theory for Economics and Finance”, Quantum Field Theory for Economics and Finance.1-10.

Montoya J,Maya C.(2016). Comparación de metodologías de optimización de carteras: markowitz vs. black-litterman, para activos financieros colombianos, Proyecto de grado para optar al título de magíster en administración financiera EAFIT. (3-10)

D-Wave.(2022). “Multiverse Computing: Optimizing Financial Portfolios with Quantum Computing”. https://www.dwavesys.com/solutions-and-products/financial-services/

Gonzáles, Saul.(2021). “Mejora De Las Modelizaciones Qubo Del Problema Del Viajante Y Enrutamiento De Vehículos Para Su Resolución Mediante Quantum Annealing”, Trabajo de grado de Matemáticas de la Universidad de Valladolid.

Abhishek Gunjan, Siddhartha Bhattacharyya. (2022). “A brief review of portfolio optimization techniques”, Artificial Intelligence Review (2023) 56:3847–3886

Franco Luis, Avendaño Claudia. (2011). “Modelo de markowitz y modelo de Black-Litterman en la optimización de Portfolios de inversión Programa de Ingeniería Financiera y de Negocios, Instituto Tecnológico Metropolitano.

Todor Stoilov, Krasimira Stoilova. (2020). “Analytical Overview and Applications of Modified Black-Litterman Model for Portfolio Optimization”. CYBERNETICS AND INFORMATION TECHNOLOGIES · Volume 20, No 2.

Besslera, Wolfgang , & Opferb, Heiko. (2017). “Multi-asset portfolio optimization and out-of-sample performance: an evaluation of Black–Litterman, mean-variance, and naïve diversification approaches”. The European Journal of Finance, 2017 Vol. 23, No. 1, 1–30, http://dx.doi.org/10.1080/1351847X.2014.953699

Rigetti Quantum Computing (2016). “Quantum Approximate Optimization Algorithm (QAOA) — Grove 1.7.0 documentation (grove-docs.readthedocs.io)”

Brandhofer Sebastian (2022). “Benchmarking the performance of portfolio optimization with QAOA”. Quantum Information Processing.

List of S&P 500 companies (2023) – Wikipedia. List of S&P 500 companies: Revision history - Wikipedia.

“Benchmarking the performance of portfolio optimization with QAOA”. Quantum Information Processing. Quantum Information Processing (2023) 22:25 https://doi.org/10.1007/s11128-022-03766-5.

Dehn Vanessa and Wellens Thomas (2023). “A hybrid quantum-classical approach to warm-starting optimization Vanessa Dehn and Thomas Wellens”. Fraunhofer Institute for Applied Solid State Physics IAF, Tullastr. 72, 79108 Freiburg, Germany.

Gomes Claudio (2022). “An Empirical Study on the Use of Quantum Computing for Financial Portfolio Optimization”. SN Computer Science (2022) 3: 335. https://doi.org/10.1007/s42979-022-01215-9.

Introduction to the Quantum Approximate Optimization Algorithm and Applications.(2021) Qiskit global summer school on quantum machine learning - 2021. https://learn.qiskit.org/summer-school/2021/lec5-2-introduction-quantum-approximate-optimization-algorithm-applications.

Frazo Duarte (2021). “Quantum Algorithms for Portfolio Management”. Master in Computing Science. Porto University.

Cohen Jeffrey , & Khan Alex. (2020). “Portfolio Optimization of 40 Stocks Using DWaves Quantum Annealer”. Chicago Quantum.

Black, F., & Litterman, R. (1992). Global portfolio optimization. Financial analysts journal, 48(5), 28-43.

Markowitz, H. (1952). Portfolio Optimization. The Journal of Finance, 7(1), 77-91.

Mari, A., Bromley, T. R., & Killoran, N. (2020). Estimating the gradient and higher-order derivatives on quantum hardware. https://doi.org/10.1103/PhysRevA