Trade a Positive-definite Portfolio

1st Runhao Li
BNU-HKBU United International College
Zhuhai 519087, China
r130034015@mail.uic.edu.cn

2nd Qing-Guo Wang
Institute of AI and Future Networks
Beijing Normal University, Zhuhai 519087, China
Guangdong Provincial Key Laboratory IRADS
Department of Computer Science, BNU-HKBU United International College
Zhuhai 519087, China
wangqingguo@bnu.edu.cn

***Abstract*—This paper is concerned with developing a portfolio optimization model that maximizes returns while ensuring non-negative wealth change, regardless of market fluctuations. The objective is to enable investors to allocate their assets in a way that guarantees non-negative wealth change over time, even as economic factors vary. We construct the portfolio using nonlinear multi-factor models and formulate the problem as a semi-definite programming (SDP) problem. By solving this problem, we derive the optimal asset allocation that maximizes asset change while ensuring it remains non-negative. In practical scenarios, we apply this model to real-market situations, specifically the US 2023 stock market, and the results show strong performance, with the model achieving good returns even under real-market conditions.**

***Index Terms*—Stock Market, Portfolio Optimization, Semidefinite Programming, Multi-Factor Models, Asset Allocation, Nonnegative Wealth Change, US Stock Market.**

I. Introduction

Portfolio management is a fundamental aspect of financial decision-making, involving the careful selection, balancing, and management of a collection of investments to achieve specific financial objectives. The primary goal of portfolio management is to maximize returns while minimizing risks, often through diversification of assets. Diversification helps spread risk by investing in a variety of asset classes, such as equities, bonds, real estate, and commodities, thereby reducing the impact of any single asset's poor performance on the overall portfolio [1]. Over time, various strategies have been developed to guide portfolio construction, ranging from passive approaches like index tracking, which aim to replicate market performance [2], to more active strategies that involve selecting specific securities based on individual forecasts, research, or technical analysis. While active strategies attempt to outperform the market by predicting price movements, they often require constant adjustments to maintain optimal positions, which can be resource-intensive and difficult to sustain over time [3].

These portfolio management strategies, however, are highly sensitive to a variety of market factors. The performance of any given portfolio is often influenced by dynamic and sometimes unpredictable forces such as interest rates, inflation, stock market indices, commodity prices, geopolitical events, and changes in investor sentiment [4], [5]. For example, an unexpected increase in interest rates may negatively affect stock prices, while a sudden drop in oil prices could impact the profitability of certain industries [6]. These factors not only cause fluctuations in asset prices but also alter the risk-return profile of different investment opportunities. As a result, traditional portfolio management approaches—whether passive or active—can struggle to consistently generate positive returns when market conditions change rapidly or unexpectedly, leading to suboptimal outcomes in many cases [7].

In light of these challenges, our objective is to develop a portfolio optimization method that can ensure profitability, regardless of fluctuations in market conditions. The goal is to create a strategy that remains robust under changing circumstances, making the portfolio resilient to external shocks. To achieve this, we propose a model based on multi-factor models that incorporate key economic variables, such as changes in stock market indices and interest rates, to explain asset returns. By considering the interrelationships between these factors and asset returns, our model enables dynamic portfolio rebalancing, which adapts to market changes while maintaining profitability. The method uses semi-definite programming (SDP) to optimize asset allocations, ensuring that the wealth change from one period to the next remains non-negative, even as the underlying market factors fluctuate. This approach maximizes wealth while ensuring that the portfolio continues to perform well, regardless of how the market conditions evolve over time.

This paper is organized as follows. In Section II, we present the theoretical foundations of our method, explaining the model and illustrating it with a simple case. In Section III, we extend the method to the general case, involving the application of semi-definite programming (SDP) for optimization. In Section IV, we discuss the results of our simulation and analysis, comparing the performance of our method with other portfolio strategies. Finally, in Section V, we conclude with a summary of the findings and potential directions for future research.

II. Motivating Portfolio

Consider n assets, s_i, $i = 1, 2, \dots, n$. Let t be the asset re-balancing time instant, $t = 0, 1, \dots, T-1$. Let $p_i(t)$ be the price of asset s_i at time t, and $u_i(t)$ be the units of asset s_i held by the investor during the time interval $[t, t+1]$. The total wealth $W(t)$ of the investor at time t is expressed [12] as

$$W(t) = \sum_{i=1}^{n} u_i(t)p_i(t), \quad t = 0, 1, \ldots, T-1, \\ i = 1, 2, \ldots, n. \tag{1}$$

For one time interval $[t, t+1]$ rebalancing, we consider the time point $t+1^-$, which represents the moment right before the reallocation occurs at time $t+1$. At this moment, the units of each asset are still $u_i(t)$, the same as they were at time t , while the price of each asset has changed from $p_i(t)$ to $p_i(t+1^-)$. Then the total wealth at the time point $t+1^-$ is given by

$$W(t+1^-) = \sum_{i=1}^{n} u_i(t)p_i(t+1^-), \quad t = 0, 1, \ldots, T-1, \\ i = 1, 2, \ldots, n. \tag{2}$$

The change in the total wealth over the period $[t, t+1^-]$ is expressed as

$$\begin{aligned} w(t, t+1^-) &= W(t+1^-) - W(t) \\ &= \sum_{i=1}^{n} u_i(t)(p_i(t+1^-) - p_i(t)) \\ &= \sum_{i=1}^{n} v_i(t)r_i(t+1) \\ &\quad t = 0, 1, \ldots, T-1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{3}$$

where $v_i(t) = p_i(t) \cdot u_i(t)$ represents the total value of the i-th stock at time t and $r_i(t+1)$ denotes the rate of return of asset s_i over the time interval $[t, t+1^-]$, defined as $\frac{p_i(t+1^-) - p_i(t)}{p_i(t)}$. Suppose price continuity, so that $p_i(t+1) = p_i(t+1^-)$. This leads to wealth continuity, $W(t+1) = W(t+1^-)$. Thus, the total wealth change over the time intervals $[t, t+1^-]$ and $[t, t+1]$ remains the same, that is, $w(t, t+1^-) = w(t, t+1)$. The expected returns of assets can be explained by multiple systematic factors [29]. To further investigate $w(t, t+1)$, we decide to employ a multi-factor model to express $r_i(t)$. [15] These factors, denoted as f_j, $j = 1, 2, \ldots, m$, represent different economic variables or indices that can influence the returns of assets. Mathematically, for an asset s_i, the rate of its return $r_i(t)$ at time t may be described by a linear multi-factor model

$$r_i(t) = a_{i0} + \sum_{j=1}^{m} a_{ij} f_j(t), \quad t = 0, 1, \ldots, T-1, \\ i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m. \tag{4}$$

To illustrate our method, we first consider a simple case of two factors $f_1(t)$, $f_2(t)$ and two assets s_1, s_2. Then, (4) becomes

$$r_i(t) = a_{i0} + a_{i1}f_1(t) + a_{i2}f_2(t), \quad t = 0, 1, \ldots, T-1. \tag{5}$$

Substituting (5) with $t+1$ into (3), we get the wealth change

$$w(t, t+1) = \sum_{i=1}^{2} v_i(t)\Big(a_{i0} + a_{i1}f_1(t+1) + a_{i2}f_2(t+1)\Big) \\ t = 0, 1, \ldots, T-1. \tag{6}$$

The total wealth change $w(t, t+1)$ is linearly influenced by the factors $f_1(t+1)$ and $f_2(t+1)$. Note the values of these factors are typically unpredictable and constantly fluctuating in the market [2], making it challenging to control the wealth change simply by adjusting the allocation of each asset. As a result, relying solely on adjusting the units $u_i(t)$ for each asset to ensure that the total wealth change $w(t, t+1)$ remains non-negative under a linear model is impossible. For example, if positive $f_1(t+1)$ and $f_2(t+1)$ make $w(t, t+1)$ positive, then negative $f_1(t+1)$ and $f_2(t+1)$ will make $w(t, t+1)$ negative. Therefore, it is necessary to consider using a nonlinear model to enable positive wealth change regardless of sign of $f_1(t+1)$ and $f_2(t+1)$. Replace (5) by the following nonlinear or quadratic multi-factor model

$$\begin{aligned} r_i(t) = a_{i0} + a_{i1}f_1(t) + a_{i2}f_2(t) + a_{i3}f_1^2(t) + \\ a_{i4}f_2^2(t) + a_{i5}f_1(t)f_2(t), \\ t = 0, 1, \ldots, T-1, \quad i = 1, 2 \end{aligned} \tag{7}$$

where a_{i0} is the constant term, and a_{ij}, $j = 1, \ldots, 5$, are coefficients. Substituting (7) with $t+1$ into equation (3) yields

$$\begin{aligned} w(t, t+1) = \sum_{i=1}^{2} v_i(t)\Big(&a_{i0} + a_{i1}f_1(t+1) + a_{i2}f_2(t+1) \\ &+ a_{i3}f_1^2(t+1) + a_{i4}f_2^2(t+1) \\ &+ a_{i5}f_1(t+1)f_2(t+1)\Big), \\ &t = 0, 1, \ldots, T-1, \quad i = 1, 2 \end{aligned} \tag{8}$$

Let

$$f(t) = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix} \tag{9}$$

$$A = \begin{bmatrix} \sum_{i=1}^{2} v_i(t)a_{i3} & \sum_{i=1}^{2} v_i(t)\frac{a_{i5}}{2} \\ \sum_{i=1}^{2} v_i(t)\frac{a_{i5}}{2} & \sum_{i=1}^{2} v_i(t)a_{i4} \end{bmatrix} \tag{10}$$

$$b = \begin{bmatrix} \sum_{i=1}^{2} v_i(t)a_{i1} \\ \sum_{i=1}^{2} v_i(t)a_{i2} \end{bmatrix} \tag{11}$$

$$c = \sum_{i=1}^{2} v_i(t)a_{i0} \tag{12}$$

Under (9)-(12), we rewrite the total wealth change (8) as a quadratic form

$$w(t, t+1) = f(t+1)^T A f(t+1) + b^T f(t+1) + c \tag{13}$$

Then we transform (13) into a pure quadratic form by defining the following vector and matrix

$$\overline{f}(t+1) = \begin{bmatrix} f(t+1) \\ 1 \end{bmatrix} \tag{14}$$

$$\mathrm{M} = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \tag{15}$$

Thus, (13) becomes

$$w(t,t+1) = \overline{f}(t+1)^T \mathrm{M} \overline{f}(t+1) \tag{16}$$

Based on this pure quadratic form, we can observe that as long as the matrix M remains positive semi-definite, the total wealth change will always be non-negative, regardless of how the factors vary. However, since the values of $\overline{f}(t+1)$ at time t are unpredictable, we want to devise our investment strategy without knowing $\overline{f}(t+1)$ to ensure that the wealth change from time t to $t+1$ is not only non-negative but also maximized. Therefore, we adjust the objective function to make it dependent solely on the stock investment amounts $v(t)$, removing the influence of the uncertain values of $\overline{f}(t+1)$. To achieve this, we consider the spectral properties of the matrix M [16]. Specifically, we use the fact that the quadratic form involving a symmetric matrix can be expressed in terms of its eigenvalues and eigenvectors. The eigenvalue decomposition of M yields

$$\mathrm{M} = \mathrm{Q}\Lambda\mathrm{Q}^\top \tag{17}$$

where $\mathrm{Q} = [\mathrm{q}_1, \mathrm{q}_2, \mathrm{q}_3]$ is an orthogonal matrix with the normalized eigenvectors, q_ℓ, of M as its columns, and Λ is a diagonal matrix containing the eigenvalues λ_ℓ of M (since M is 3×3 in this case, $\ell = 1, 2, 3$). Substituting this decomposition into the quadratic form, we have

$$\begin{aligned} w(t,t+1) &= \overline{f}(t+1)^\top \mathrm{M} \overline{f}(t+1) \\ &= \overline{f}(t+1)^\top \mathrm{Q}\Lambda\mathrm{Q}^\top \overline{f}(t+1) \\ &= (\mathrm{Q}^\top \overline{f}(t+1))^\top \Lambda (\mathrm{Q}^\top \overline{f}(t+1)) \\ &= \sum_{\ell=1}^{3} \lambda_\ell \left(\mathrm{q}_\ell^\top \overline{f}(t+1)\right)^2, \quad t = 0, 1, \ldots, T-1, \end{aligned} \tag{18}$$

For a positive semidefinite matrix M, all its eigenvalues λ_ℓ are non-negative. This means that $w(t,t+1)$, expressed as a weighted sum of the non-negative terms $\left(\mathrm{q}_\ell^\top \overline{f}(t+1)\right)^2$, is also non-negative. Larger eigenvalues λ_ℓ yield a higher value of $w(t,t+1)$. Thus, maximizing the sum of the eigenvalues approximates maximizing $w(t,t+1)$ when $\overline{f}(t+1)$ is unknown. Therefore, we choose the sum of the eigenvalues as the objective function and maximize it. Importantly, the sum of the eigenvalues of M is equal to its trace [19],

$$\sum_{\ell=1}^{3} \lambda_\ell = \mathrm{trace}(\mathrm{M}). \tag{19}$$

Thus, we reformulate our optimization problem to maximize the trace of M, while ensuring that the matrix M must be positive semidefinite. Additionally, the sum of the investments $v_i(t)$ should equal the total wealth at time t, ensuring that all available wealth is allocated. The problem is stated as

$$\begin{aligned} \max_{v(t)} \quad & \mathrm{trace}(\mathrm{M}) \\ \text{s.t.} \quad & \mathrm{M} \succeq 0, \\ & \sum_{i=1}^{n} v_i(t) = W(t). \end{aligned} \tag{20}$$

where $v(t) = \begin{bmatrix} v_1(t) & v_2(t) \end{bmatrix}$. To ensure that the matrix M is positive semi-definite, the following conditions must be satisfied [18], [19],

$$A_{11} = \sum_{i=1}^{2} v_i(t) a_{i3} \geq 0, \tag{21}$$

$$\det(A) = \left(\sum_{i=1}^{2} v_i(t) a_{i3}\right)\left(\sum_{i=1}^{2} v_i(t) a_{i4}\right) - \left(\sum_{i=1}^{2} v_i(t) \frac{a_{i5}}{2}\right)^2 \geq 0, \tag{22}$$

$$c - b^T A^{-1} b \geq 0, \tag{23}$$

where A_{11} is the top-left element of the matrix A, $\det(A)$ is the determinant of the 2×2 submatrix A, and $c - b^T A^{-1} b$ represents the Schur complement of M. Then, the problem (20) is restated as P1:

$$\begin{aligned} \max_{v(t)} \quad & G = \sum_{i=1}^{2} v_i(t) a_{i3} + \sum_{i=1}^{2} v_i(t) a_{i4} + \sum_{i=1}^{2} v_i(t) a_{i0} \\ \text{s.t.} \quad & \\ & \sum_{i=1}^{2} v_i(t) a_{i3} \geq 0, \\ & \left(\sum_{i=1}^{2} v_i(t) a_{i3}\right)\left(\sum_{i=1}^{2} v_i(t) a_{i4}\right) - \left(\sum_{i=1}^{2} v_i(t) \frac{a_{i5}}{2}\right)^2 \geq 0, \\ & c - b^\top A'^{-1} b \geq 0, \\ & \sum_{i=1}^{2} v_i(t) = W(t), \end{aligned} \tag{24}$$

We use the Lagrangian method [20] to solve P1 for the optimal $v_1^*(t)$ and $v_2^*(t)$. The Lagrangian function for our optimization problem is given by

$$\begin{aligned} \mathcal{L}(v_1(t), v_2(t), \lambda_1, \lambda_2, \lambda_3, \lambda_4) &= \sum_{i=1}^{2} v_i(t) a_{i3} + \sum_{i=1}^{2} v_i(t) a_{i4} + \sum_{i=1}^{2} v_i(t) a_{i0} \\ & - \lambda_1 \Big(\sum_{i=1}^{n} v_i(t) - W(t)\Big) - \lambda_2 \Big(c - b^T A^{-1} b\Big) \\ & - \lambda_3 \Big(\sum_{i=1}^{2} v_i(t) a_{i3}\Big) - \lambda_4 \Big(\Big(\sum_{i=1}^{2} v_i(t) a_{i3}\Big)\Big(\sum_{i=1}^{2} v_i(t) a_{i4}\Big) - \\ & \Big(\sum_{i=1}^{2} v_i(t) \frac{a_{i5}}{2}\Big)^2\Big). \end{aligned} \tag{25}$$

where λ_1 is the Lagrange multiplier for the wealth constraint $\sum_{i=1}^{n} v_i(t) = W(t)$, λ_2 is the Lagrange multiplier for the Schur complement condition $c - b^T A^{-1} b \geq 0$, λ_3 is the Lagrange multiplier for the non-negativity constraint

$\sum_{i=1}^2 v_i(t)a_{i3} \geq 0$, and λ_4 is the Lagrange multiplier for the determinant condition $\left(\sum_{i=1}^2 v_i(t)a_{i3}\right)\left(\sum_{i=1}^2 v_i(t)a_{i4}\right) - \left(\sum_{i=1}^2 v_i(t)\frac{a_{i5}}{2}\right)^2 \geq 0$. To find the optimal $v_1^*(t)$, $v_2^*(t)$, we first address the equality constraints by setting the partial derivatives of the Lagrangian with respect to $v_1(t)$, $v_2(t)$, and λ_1 equal to zero. This yields the following conditions

$$\frac{\partial \mathcal{L}}{\partial v_1(t)} = a_{13} + a_{14} + a_{10} - \lambda_1 = 0, \tag{26}$$

$$\frac{\partial \mathcal{L}}{\partial v_2(t)} = a_{23} + a_{24} + a_{20} - \lambda_1 = 0, \tag{27}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = \sum_{i=1}^{n} v_i(t) - W(t) = 0. \tag{28}$$

For the inequality constraints, we apply the Karush-Kuhn-Tucker (KKT) conditions [21], which require the following complementary slackness conditions to hold:

$$\lambda_2 \geq 0, \quad c' - b'^T A'^{-1} b' \geq 0, \quad \lambda_2\left(c' - b'^T A'^{-1} b'\right) = 0 \tag{29}$$

$$\lambda_3 \geq 0, \quad \sum_{i=1}^{2} v_i(t)a_{i3} \geq 0, \quad \lambda_3\left(\sum_{i=1}^{2} v_i(t)a_{i3}\right) = 0 \tag{30}$$

$$\begin{gathered}\lambda_4 \geq 0, \quad \left(\sum_{i=1}^{2} v_i(t)a_{i3}\right)\left(\sum_{i=1}^{2} v_i(t)a_{i4}\right) - \left(\sum_{i=1}^{2} v_i(t)\frac{a_{i5}}{2}\right)^2 \geq 0, \\ \lambda_4\left(\left(\sum_{i=1}^{2} v_i(t)a_{i3}\right)\left(\sum_{i=1}^{2} v_i(t)a_{i4}\right) - \left(\sum_{i=1}^{2} v_i(t)\frac{a_{i5}}{2}\right)^2\right) = 0.\end{gathered} \tag{31}$$

To solve the system of equations from (26)-(31) for $v_1^*(t)$ and $v_2^*(t)$, we examine multiple configurations based on whether each constraint is active (binding) or inactive (non-binding). Each configuration introduces a distinct set of conditions to the Lagrangian function, guiding the steps required to find $v_1(t)$ and $v_2(t)$.

In case 1, all constraints are active. Here, the non-negativity constraint $\sum_{i=1}^2 v_i(t)a_{i3} = 0$, the determinant condition $\left(\sum_{i=1}^2 v_i(t)a_{i3}\right)\left(\sum_{i=1}^2 v_i(t)a_{i4}\right) - \left(\sum_{i=1}^2 v_i(t)\frac{a_{i5}}{2}\right)^2 = 0$, and the Schur complement condition $c - b^\top A'^{-1} b = 0$ are all binding. In this case, we incorporate each constraint directly as an equality in the Lagrangian. By doing so, we simplify the system by eliminating terms associated with these constraints and focus solely on the reduced system of partial derivatives with respect to $v_1(t)$, $v_2(t)$, and λ. Under this setup, we derive the following solutions for $v_1(t)$ and $v_2(t)$:

$$v_1^*(t) = \frac{a_{23} + a_{24} + a_{20}}{a_{13} + a_{14} + a_{10} + a_{23} + a_{24} + a_{20}} W(t),$$

$$v_2^*(t) = \frac{a_{13} + a_{14} + a_{10}}{a_{13} + a_{14} + a_{10} + a_{23} + a_{24} + a_{20}} W(t).$$

These results show the optimal allocation values under this configuration, allowing for wealth maximization while ensuring non-negative wealth change over time. Due to space limitations, we will not solve the remaining cases in full detail.

In case 2, the non-negativity and determinant conditions are active, while the Schur complement condition is inactive. Here, the Schur complement condition $c - b^\top A'^{-1} b > 0$ holds as a strict inequality, implying that $\lambda_2 = 0$ by the KKT conditions. We treat the non-negativity constraint and the determinant condition as equalities. These are incorporated into the Lagrangian, enabling us to solve the system by focusing only on the active constraints.

In case 3, the Schur complement and determinant conditions are active, while the non-negativity constraint is inactive. In this case, the non-negativity constraint is satisfied as $\sum_{i=1}^2 v_i(t)a_{i3} > 0$, which leads to $\lambda_3 = 0$ by complementary slackness. Here, we enforce the Schur complement and determinant conditions as equalities in the Lagrangian, allowing us to focus on solving only these two active constraints, which simplifies the partial derivatives and yields the feasible values for $v_1(t)$ and $v_2(t)$.

In case 4, only the determinant condition is active, with both the Schur complement condition and the non-negativity constraint inactive. Here, $c - b^\top A'^{-1} b > 0$ and $\sum_{i=1}^2 v_i(t)a_{i3} > 0$, which results in $\lambda_2 = 0$ and $\lambda_3 = 0$. This leaves only the determinant condition as an equality constraint. We incorporate this into the Lagrangian as an equality and proceed by solving the remaining partial derivatives with respect to $v_1(t)$ and $v_2(t)$, focusing solely on this active constraint.

By reallocating the investment according to the optimal $v_1^*(t)$ and $v_2^*(t)$ solved from our system of equations, we can achieve our goal of maximizing wealth change while ensuring that it remains non-negative over the time period from t to $t+1$ and is unaffected by the uncertainty of $\overline{f}(t+1)$, which demonstrates that the nonlinear model has been effective in achieving our goal.

In real-world environments, as the number of stocks increases, the optimization method can no longer rely on simple differentiation, but instead requires more sophisticated approaches. In the next section, we will present the general case with the numerical optimization.

III. General portfolio

In the general case, we consider n assets, s_i, $i = 1, 2, \ldots, n$, and m factors, $f_j(t)$, $j = 1, 2, \ldots, m$, and the multi-factor model still adopts the similar nonlinear form as described in (7). The model is formulated as follows

$$\begin{gathered} r_i(t) = a_{i0} + \sum_{j=1}^{m} a_{ij} f_j(t) + \sum_{1 \leq j < J \leq m} a_{ijJ} f_j(t) f_J(t), \\ t = 0, 1, \ldots, T-1, \quad i = 1, 2, \ldots, n, j = 1, 2, \ldots, m. \end{gathered} \tag{32}$$

Substituting (32) with $t+1$ into equation (3) yields

$$\begin{aligned} w(t, t+1) = \sum_{i=1}^{n} v_i(t)\Big(&a_{i0} + \sum_{j=1}^{m} a_{ij} f_j(t+1) \\ &+ \sum_{1 \leq j < J \leq m} a_{ijJ} f_j(t+1) f_J(t+1)\Big), \\ &t = 0, 1, \ldots, T-1, \quad i = 1, 2, \ldots, n, j = 1, 2, \ldots, m. \end{aligned} \tag{33}$$

Define the following vector,

$$f(t)=\begin{bmatrix} f_1(t)\\ f_2(t)\\ \vdots\\ f_m(t)\end{bmatrix}. \tag{34}$$

$$\mathrm{A}_i=\begin{bmatrix} a_{i11} & \frac{a_{i12}}{2} & \cdots & \frac{a_{i1m}}{2}\\ \frac{a_{i12}}{2} & a_{i22} & \cdots & \frac{a_{i2m}}{2}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{a_{i1m}}{2} & \frac{a_{i2m}}{2} & \cdots & a_{imm}\end{bmatrix} \tag{35}$$

$$\mathrm{b}_i=\begin{bmatrix} a_{i1}\\ a_{i2}\\ \vdots\\ a_{im}\end{bmatrix} \tag{36}$$

$$A=\sum_{i=1}^{n} A_i v_i(t), \tag{37}$$

$$b=\sum_{i=1}^{n} b_i v_i(t), \tag{38}$$

$$c=\sum_{i=1}^{n} a_{i0} v_i(t), \tag{39}$$

where $t=0,1,\ldots,T-1$ and $i=1,2,\ldots,n$. With (34)-(39), $w(t,t+1)$, the wealth change during time period $[t,t+1]$ is expressed as

$$w(t,t+1)=f(t+1)^{\top}Af(t+1)+b^{\top}f(t+1)+c. \tag{40}$$

To transform (40) into a quadratic form, we define the following vector

$$\overline{f}(t+1)=\begin{bmatrix} f(t+1)\\ 1\end{bmatrix} \tag{41}$$

$$\mathrm{M}=\begin{bmatrix} A & \frac{1}{2}b\\ \frac{1}{2}b^{\top} & c\end{bmatrix} \tag{42}$$

Then (40) becomes a pure quadratic form,

$$w(t,t+1)=\overline{f}(t+1)^{\top}\mathrm{M}\overline{f}(t+1) \tag{43}$$

We aim to ensure that $w(t,t+1)$ is non-negative and maximized, irrespective of the sign and value of $\overline{f}(t+1)$. We formulate this problem as

$$\begin{aligned} \max_{v(t)} \quad & \mathrm{trace}(\mathrm{M})\\ \text{s.t.} \quad & \mathrm{M}\succeq 0,\\ & \sum_{i=1}^{n} v_i(t)=W(t). \end{aligned} \tag{44}$$

where $v(t)=\begin{bmatrix} v_1(t) & v_2(t) & \ldots & v_n(t)\end{bmatrix}$, we calculate $\mathrm{trace}(\mathrm{M})$ as

$$\begin{aligned} \mathrm{trace}(\mathrm{M}) &= \mathrm{trace}(\mathrm{A})+c\\ &=\sum_{i=1}^{n}\left(a_{i11}+a_{i22}+\ldots+a_{imm}\right)v_i(t)+\sum_{i=1}^{n}a_{i0}v_i(t)\\ &=\sum_{i=1}^{n}\left(a_{i0}+a_{i11}+a_{i22}+\ldots+a_{imm}\right)v_i(t). \end{aligned} \tag{45}$$

$$t=0,1,\ldots,T-1, \quad i=1,2,\ldots,n.$$

Then (44) is rewritten as

$$\begin{aligned} \max_{\mathrm{v}(t)} \quad & \sum_{i=1}^{n}\left(a_{i0}+a_{i11}+a_{i22}+\ldots+a_{imm}\right)v_i(t)\\ \text{s.t.} \quad & \mathrm{M}\succeq 0,\\ & \sum_{i=1}^{n} v_i(t)=W(t). \end{aligned} \tag{46}$$

The objective function is now linear in $v_i(t)$, as seen in (46), and the constraints include a positive semidefinite matrix inequality. This structure aligns with the standard form of Semidefinite Programming (SDP). Therefore, we consider using SDP techniques for subsequent optimization to efficiently determine the optimal investment strategy.

The general form of a Semidefinite Programming (SDP) problem is written [22] as follows,

$$\begin{aligned} \min_{\mathrm{x}} \quad & \mathrm{d}^{\top}\mathrm{x}\\ \text{s.t.} \quad & \mathrm{G}(\mathrm{x})=\mathrm{G}_0+\sum_{i=1}^{n}x_i\mathrm{G}_i\succeq 0,\\ & \mathrm{U}\mathrm{x}=\eta \end{aligned} \tag{47}$$

where $\mathrm{G}(\mathrm{x})\succeq 0$ represents the positive semidefinite constraint, and $\mathrm{U}\mathrm{x}=\eta$ represents the equality constraint. To solve this SDP problem, we employ the interior point method [25], which transforms the constrained optimization problem into unconstrained problems by incorporating a barrier term into the objective function. We reformulate the SDP problem corresponding to (47) as

$$\begin{aligned} \min_{\mathrm{x}} \quad & \mathrm{d}^{\top}\mathrm{x}-\mu\log\det\left(\mathrm{G}(\mathrm{x})\right)\\ \text{s.t.} \quad & \mathrm{U}\mathrm{x}=\eta, \end{aligned} \tag{48}$$

where $\mu>0$ is the barrier parameter, and $\log\det\left(\mathrm{G}(\mathrm{x})\right)$ is the logarithmic barrier function associated with the positive semidefinite constraint $\mathrm{G}(\mathrm{x})\succeq 0$. As μ approaches zero, the solution obtained from the interior point method converges to the solution of the original SDP. To solve (48), we consider the Lagrangian function

$$\mathcal{L}(\mathrm{x},\lambda)=\mathrm{d}^{\top}\mathrm{x}-\mu\log\det\left(\mathrm{G}(\mathrm{x})\right)+\lambda^{\top}(\mathrm{U}\mathrm{x}-\eta) \tag{49}$$

where λ is the vector of Lagrange multipliers associated with the equality constraints. The optimality conditions are obtained by ensuring that the following equations [25] hold

$$\begin{aligned}\nabla_{\mathrm{x}}\mathcal{L}(\mathrm{x},\lambda) &= \mathrm{d} - \mu\nabla_{\mathrm{x}}\log\det\left(\mathrm{G}(\mathrm{x})\right) + \mathrm{U}^\top\lambda = 0\\ \nabla_{\lambda}\mathcal{L}(\mathrm{x},\lambda) &= \mathrm{U}\mathrm{x} - \eta = 0\end{aligned} \tag{50}$$

To solve the nonlinear equations given by (50), we apply Newton's method to update x, while λ is updated using a linear update approach. The updates are as follows

$$\begin{aligned}\mathrm{x}_{k+1} &= \mathrm{x}_k - \left(\nabla^2_{\mathrm{x}}\mathcal{L}(\mathrm{x},\lambda)\right)^{-1}\left(\mathrm{d} - \mu_k\nabla_{\mathrm{x}}\log\det\left(\mathrm{G}(\mathrm{x})\right) + \mathrm{U}^\top\lambda\right)\\ \lambda_{k+1} &= \lambda_k - (\mathrm{U}\mathrm{x}_k - \eta)\end{aligned} \tag{51}$$

Then we decrease the barrier parameter μ using an update rule as

$$\mu_{k+1} = \sigma\cdot\mu_k \tag{52}$$

where $\sigma \in (0,1)$ controls the rate of decrease. By progressively reducing μ and refining the solution, we ensure that the iterates approach feasibility with respect to the original constraints. Following each iteration, we check for convergence by evaluating whether the norm of the gradients $\|\nabla_{\mathrm{x}}\mathcal{L}(\mathrm{x}_{k+1},\lambda_{k+1})\|$ is smaller than a predefined threshold β (e.g., $\beta = 10^{-6}$), and the feasibility of the equality constraint $\mathrm{U}\mathrm{x}_{k+1} = \mathrm{n}$. We will proceed to the next iteration until these convergence criteria are met [11]. Once the method converges, the obtained x^* represents the optimal solution, which satisfies the necessary optimality conditions and constraints for the decision variables.

In our problem, we transform (46) into the form of (47) by utilizing (35)-(39). As a result, we obtain

$$\begin{aligned}\min_{v(t)}\quad & -\begin{bmatrix} a_{10}+a_{111}+a_{122}+\ldots+a_{1mm}\\ a_{20}+a_{211}+a_{222}+\ldots+a_{2mm}\\ \vdots\\ a_{n0}+a_{n11}+a_{n22}+\ldots+a_{nmm}\end{bmatrix}^\top \mathrm{v}(t)\\ \text{s.t.}\quad & \sum_{i=1}^{n} v_i(t)\begin{bmatrix}\mathrm{A}_i & \frac{1}{2}\mathrm{b}_i\\ \frac{1}{2}\mathrm{b}_i^\top & a_{i0}\end{bmatrix}\succeq 0,\\ & \sum_{i=1}^{n} v_i(t) = W(t),\\ & t = 0,1,\ldots,T-1,\quad i = 1,2,\ldots,n.\end{aligned} \tag{53}$$

Then, according to (49), we obtain

$$\begin{aligned}\mathcal{L}(\mathrm{v}(t),\lambda) = &-\begin{bmatrix} a_{10}+a_{111}+a_{122}+\ldots+a_{1mm}\\ a_{20}+a_{211}+a_{222}+\ldots+a_{2mm}\\ \vdots\\ a_{n0}+a_{n11}+a_{n22}+\ldots+a_{nmm}\end{bmatrix}^\top \mathrm{v}(t)\\ &-\mu\log\det\left(\sum_{i=1}^{n} v_i(t)\begin{bmatrix}\mathrm{A}_i & \frac{1}{2}\mathrm{b}_i\\ \frac{1}{2}\mathrm{b}_i^\top & a_{i0}\end{bmatrix}\right)\\ &+\lambda\left(\sum_{i=1}^{n} v_i(t) - W(t)\right),\\ & t = 0,1,\ldots,T-1,\quad i = 1,2,\ldots,n.\end{aligned} \tag{54}$$

To find the optimal solution of (54), we use the method in (51) and (52)

$$\begin{aligned}\mathrm{v}_{k+1}(t) = \mathrm{v}_k(t) - \left(\nabla^2_{\mathrm{v}(t)}\mathcal{L}(\mathrm{v}(t),\lambda_k)\right)^{-1}&\left(-\begin{bmatrix} a_{10}+a_{111}+a_{122}+\ldots+a_{1mm}\\ a_{20}+a_{211}+a_{222}+\ldots+a_{2mm}\\ \vdots\\ a_{n0}+a_{n11}+a_{n22}+\ldots+a_{nmm}\end{bmatrix}\right.\\ &\left.+\mu_k\nabla_{\mathrm{v}(t)}\log\det\left(\sum_{i=1}^{n} v_i(t)\begin{bmatrix}\mathrm{A}_i & \frac{1}{2}\mathrm{b}_i\\ \frac{1}{2}\mathrm{b}_i^\top & a_{i0}\end{bmatrix}\right)\right),\\ \lambda_{k+1} = \lambda_k - \left(\sum_{i=1}^{n} v_i(t) - W(t)\right),&\\ \mu_{k+1} = \sigma\cdot\mu_k.&\end{aligned} \tag{55}$$

Once the norm of the gradients $\|\nabla_{\mathrm{x}}\mathcal{L}(\mathrm{x}_{k+1},\lambda_{k+1})\|$ is smaller than the threshold β we set, which indicates that the function has converged, the optimal $v(t)^*$ will be obtained as the result of the optimization. The investment decisions is made based on this optimal $v(t)^*$. The following algorithm formalizes this process, detailing the steps to obtain the optimal investment allocation $v(t)^*$ through an interior point method with a barrier function.

Algorithm 1 Interior Point Method for Optimizing Investment Allocation via Barrier Function

---

Require: Coefficients a_{ij} for $i = 1,2,\ldots,n$, $j = 1,2,\ldots,m$, initial $\mathrm{v}_0(t)$, λ_0, current total wealth $W(t)$, barrier parameter $\mu_0 > 0$, decay factor $\sigma \in (0,1)$ and a small value β as the threshold of convergence.

Ensure: Optimal investment allocation $\mathrm{v}(t)^*$.

Step 1: Define the Lagrangian function $\mathcal{L}(\mathrm{v}(t),\lambda)$:

$$\begin{aligned}\mathcal{L}(\mathrm{v}(t),\lambda) = &-\mathrm{m}^\top\mathrm{v}(t) - \mu\log\det\left(\sum_{i=1}^{n} v_i(t)\mathrm{M}_i\right)\\ &+\lambda\left(\sum_{i=1}^{n} v_i(t) - W(t)\right)\end{aligned}$$

where:

$$\mathrm{m} = \begin{bmatrix} a_{10}+a_{111}+a_{122}+\ldots+a_{1mm}\\ a_{20}+a_{211}+a_{222}+\ldots+a_{2mm}\\ \vdots\\ a_{n0}+a_{n11}+a_{n22}+\ldots+a_{nmm}\end{bmatrix},\quad \mathrm{M}_i = \begin{bmatrix}\mathrm{A}_i & \frac{1}{2}\mathrm{b}_i\\ \frac{1}{2}\mathrm{b}_i^\top & a_{i0}\end{bmatrix}$$

Step 2: Newton's method iterations: for $k = 1$ to MaxIterations **do**

- Compute the gradient of the Lagrangian:

$$\nabla_{\mathrm{v}(t)}\mathcal{L} = -\mathrm{m} + \mu_k\nabla_{\mathrm{v}(t)}\log\det\left(\sum_{i=1}^{n} v_i(t)\mathrm{M}_i\right) + \lambda_k$$

- Compute the Hessian matrix of the Lagrangian:

$$\nabla^2_{\mathrm{v}(t)}\mathcal{L} = \mu_k\nabla^2_{\mathrm{v}(t)}\log\det\left(\sum_{i=1}^{n} v_i(t)\mathrm{M}_i\right)$$

- Update $\mathrm{v}(t)$ using the Newton's method step:

$$\mathrm{v}_{k+1}(t) = \mathrm{v}_k(t) - \left(\nabla^2_{\mathrm{v}(t)}\mathcal{L}\right)^{-1}\nabla_{\mathrm{v}(t)}\mathcal{L}$$

- Update the Lagrange multiplier λ:

$$\lambda_{k+1} = \lambda_k - \left(\sum_{i=1}^{n} v_i(t) - W(t)\right)$$

- Update the barrier parameter μ:

$$\mu_{k+1} = \sigma\cdot\mu_k$$

- Convergence check: **if** $\|\nabla_{\mathrm{v}(t)}\mathcal{L}\| < \beta$ (threshold) **then Terminate the loop.**
end if

end for

Step 3: Output the optimal solution $\mathrm{v}(t)^*$**:** Return $\mathrm{v}(t)^*$ and λ.

IV. Simulation

In this section, the theory developed in the preceding section is applied to a practical scenario. Using monthly data sourced from Yahoo Finance, we select 10 assets from the market of the United States in 2023, which are AMZN, GOOGL, IBM, INTC, NVDA, MSFT, TSLA, CRM, META, and QCOM. We will refer to these assets as $\{s_1, s_2, s_3, \ldots, s_{10}\}$. The data covers the prices of these assets from January 1, 2023, to November 30, 2023.

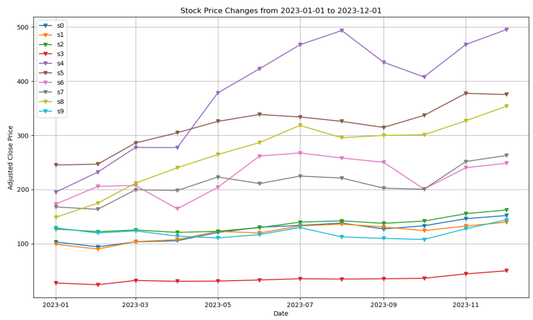

Fig. 1: Stock Prices from January 1, 2023, to November 30, 2023

We set $T = 12$, representing 12 months. The initial wealth, $W(0)$, is set to 1, representing our initial total asset value. Then, we set the number of factors, m, in (32) to 2, and construct our multi-factor model. The selection of factors is based on extensive research that has demonstrated the correlations between the rates of change in the stock index and the market interest rate with the rate of return of stocks. For instance, Smith [13] identified a strong correlation between changes in the stock index and stock returns, indicating that an increase in the stock index often results in higher stock returns. Similarly, Johnson [14] showed that fluctuations in market interest rates have a measurable impact on stock returns, with higher interest rates generally associated with lower stock returns. Furthermore, Wang [15] emphasized that both the changes in the stock index and interest rates, along with their interactions, are critical factors influencing stock returns. Based on these findings, it is plausible that the rate of return of an asset is related to the rates of change in the stock index and the market interest rate. Therefore, we denote the monthly rates of change in the stock index and the market interest rate as $f_1(t)$ and $f_2(t)$, respectively.

$$f_1(t) = \frac{s(t) - s(t-1)}{s(t-1)}, \quad t = 0, 1, \ldots, T-1. \tag{56}$$

$$f_2(t) = \frac{i(t) - i(t-1)}{i(t-1)}, \quad t = 0, 1, \ldots, T-1. \tag{57}$$

We select the S&P 500 Index (GSPC) as our stock index and the U.S. 3-Month Treasury Bill Rate (IRX) as our market interest rate. According to (56) and (57), we calculate their monthly rates of change, which are used as the factors $f_1(t)$ and $f_2(t)$ in the multi-factor model. According to (32), we construct the required multi-factor model based on the monthly returns of these 10 stocks from 2019 to 2022, and their relationship with the monthly changes in the stock index and market interest rate, namely, $f_1(t)$ and $f_2(t)$, during this period. This allows us to obtain the multi-factor models that represent the relationship between the monthly return rate and the factors we choose. Here is the algorithm to obtain the multi-factor model.

Algorithm 2 Multi-Factor Model Construction

Require: Historical monthly data for stock returns $\{s_1, s_2, \ldots, s_{10}\}$, stock index $s(t)$, and market interest rate $i(t)$ for a period of 2019–2022.

Ensure: Multi-factor model coefficients $\{a_{i0}, a_{i1}, \ldots, a_{i22}, a_{i12}\}$, $i = 1, 2, \ldots, 10$

1: Initialize the stock returns and factor variables for the period.

2: Calculate the monthly rate of change for stock index $f_1(t)$ and market interest rate $f_2(t)$ using:

$$f_1(t) = \frac{s(t) - s(t-1)}{s(t-1)}$$

$$f_2(t) = \frac{i(t) - i(t-1)}{i(t-1)}$$

3: For each stock s_i, construct the multi-factor model to describe the return rate $r_i(t)$:

$$r_i(t) = a_{i0} + a_{i1} f_1(t) + a_{i2} f_2(t) + a_{i11} f_1^2(t) + a_{i22} f_2^2(t) + a_{i12} f_1(t) f_2(t)$$

4: Use historical data from 2019–2022 to fit the multi-factor model for each stock s_i using regression, obtaining the coefficients $\mathrm{C}_i = \{a_{i0}, a_{i1}, a_{i2}, a_{i11}, a_{i22}, a_{i12}\}, i = 1, 2, \ldots, 10$.

5: Store the coefficients in a matrix C where each row corresponds to a stock, and each column represents coefficients $a_{i0}, a_{i1}, \ldots, a_{i12}$.

6: Return the matrix C.

To implement this algorithm in Python, we utilized several essential libraries to streamline the process. Pandas was used to handle and preprocess historical data, including calculating the monthly rates of change for both the stock index and market interest rate. NumPy supported the matrix operations required for efficient data manipulation and processing. We calculated the returns for each asset, as well as the monthly changes in both the stock index and market interest rate, using Pandas' built-in functions for rate of change calculations. Additionally, Statsmodels was employed to perform the regression analysis needed to construct the multi-factor model by fitting the historical returns data against the selected factors. This combination of libraries enabled us to efficiently manage the data and accurately model the relationships between asset returns and the chosen factors. Below is the coefficient matrix C for all models obtained through this process, along with their corresponding stocks.

TABLE I: Multi-factor model coefficients for 10 stocks

	a_{i0}	a_{i1}	a_{i2}	a_{i11}	a_{i22}	a_{i12}
s_1	0.0218	0.0002	-0.2482	-0.0001	1.2099	-0.0006
s_2	0.0403	0.0003	-0.3768	-0.0002	1.2650	0.0020
s_3	0.0274	0.0002	0.1018	-0.0001	-0.8144	0.0006
s_4	0.0732	0.0002	-1.2579	0.0001	2.6456	0.0024
s_5	0.0441	0.0005	0.7772	0.0001	-0.2136	-0.0093
s_6	0.0134	0.0000	-0.4317	0.0001	1.9272	-0.0011
s_7	-0.0644	0.0003	1.1494	0.0000	-1.5739	-0.0117
s_8	0.0088	0.0003	-0.2061	0.0000	1.1260	-0.0012
s_9	0.0705	0.0002	-0.1117	-0.0009	1.0529	-0.0019
s_{10}	-0.0017	0.0004	-0.0415	0.0008	-0.7364	0.0022

By using the coefficients a_{ij} from the multi-factor model, we can formulate and solve our optimization problem through Semi-Definite Programming (SDP). These coefficients capture the relationship between asset returns and underlying factors, enabling the optimization of expected returns while maintaining constraints on allocation, risk, and diversification. The process outlined in equations (53) to (55) allows us to maximize returns while adhering to these constraints. The steps of the optimization procedure—including defining the Lagrangian, applying Newton's method for refinement, and updating the Lagrange multipliers—are provided in the algorithm 1. By iterating through this process, we refine the portfolio until the optimal allocation is reached, ensuring a balanced approach to maximizing returns while managing risk effectively. To implement the optimization algorithm 1, we used NumPy to handle matrix operations and assemble the objective function. The coefficients obtained from the multi-factor model, along with the investment allocation variables $v(t)$, were utilized to construct the objective function. For solving the optimization problem, we employed CVXPY to formulate the Semidefinite Programming (SDP) problem. The objective function was set as the target to maximize while ensuring the matrix remains positive semidefinite and respecting the constraints on the investment allocations. We specifically used the MOSEK solver in CVXPY, which is an interior-point method solver [24]. This solver handled the iterative optimization process, ensuring that both the objective function and the constraints were properly maintained throughout, leading to efficient and accurate convergence of the solution. Using the optimal stock allocation $v(t)^*$ obtained through the MOSEK solver in CVXPY, we proceeded to test the theoretical performance of our portfolio. At $t = 0$, with an initial asset value $W(0) = 0$, we computed the optimal allocation $v(t)^*$ as

TABLE II: Investment Allocation at $t = 0$

Stock	Investment Allocation
AMZN	0.09
GOOGL	0.04
IBM	0.03
INTC	0.02
NVDA	0.69
MSFT	0.05
TSLA	0.03
CRM	0.02
META	0.01
QCOM	0.02

In the subsequent time periods, we will continue to use our optimization model to determine the optimal $v(t)^*$ at each time step, ensuring effective reallocation of assets based on the updated optimal values. To evaluate the performance of our optimization method, we analyze it under both theoretical conditions and practical verification, while also including a benchmark for comparison. Under theoretical condition, based on (33), the asset change $w(t, t+1)$ is calculated by using the optimal allocation $v(t)^*$ derived from our model, along with the values of two factors. Each stock's investment amount is determined based on this optimal allocation. We then add the computed wealth change to the initial asset value, updating it monthly, thereby generating the theoretical asset trajectory over time. In the actual verification, we apply the theoretical model to real-world stock prices and returns. Here, the portfolio is adjusted monthly based on actual data from the entire year of 2023, directly following the optimal allocation $v(t)^*$ from the model. Unlike in the theoretical case, where wealth change is calculated using factors, here the portfolio's value evolves naturally over time as it follows each reallocation strategy provided by the optimization model. This approach allows us to effectively evaluate the model's performance in a real-world setting, giving insight into its actual applicability. Finally, as a benchmark, we construct an equal-weighted portfolio in which an equal amount of capital is invested in each stock. Like the practical case, this benchmark is also applied in a real-market environment, allowing for a clear comparison with our optimized model and providing a basis to verify whether our model yields superior performance. The following figure illustrates the portfolio value changes over time for each of these three scenarios.

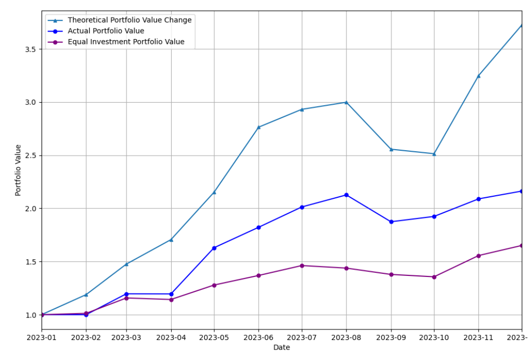

Fig. 2: Wealth of the Portfolio Over Time

In the comparison, we observe that the theoretical case shows the highest portfolio value growth. This is expected, as it represents an idealized scenario with assumptions of perfect foresight and no market frictions, providing an upper bound on potential returns. In the actual case, where the model is applied to real market data, the portfolio growth is understandably lower than in the idealized theoretical scenario, likely due to factors such as market volatility and unpredicted fluctuations. Nevertheless, the actual case still demonstrates a strong return, reaching approximately 120% by the end of the year. This is notably higher than the equal-weighted benchmark, indicating that the optimization model performs well even under real market conditions. To further validate the effectiveness of our model, we compare each strategy using additional performance metrics, including return rate, standard deviation, Sharpe ratio, and maximum drawdown. A table with these metrics is provided below.

TABLE III: Performance Metrics for Theoretical, Actual, and Equal-Weighted Cases

Metric	Theoretical Case	Actual Case	Equal-Weighted Case
Return Rate (%)	251.43	116.27	65.10
Standard Deviation (%)	46.43	11.82	22.44
Sharpe Ratio	5.42	10.58	2.90
Maximum Drawdown (%)	16.15	11.87	7.22

In the theoretical case, we observe an exceptionally high return rate of 251.43%. This result highlights the potential maximum returns under ideal conditions, where market behavior perfectly aligns with the model's predictive factors. However, this high return comes with increased volatility, indicated by a standard deviation of 46.43% and a maximum drawdown of 16.15%. The Sharpe ratio of 5.42 underscores the strong performance in this idealized scenario but also reflects the model's sensitivity to market fluctuations. Overall, the theoretical case suggests substantial profit potential, but it also highlights the risks associated with relying solely on theoretical assumptions without accounting for real market dynamics. In the actual case, we apply the model to actual market data from 2023. Here, the return rate decreases to 116.27%, as expected, since the actual model is subject to real-world fluctuations rather than the idealized conditions of the theoretical scenario. However, the actual model shows significantly improved stability, with a standard deviation of 11.82%, demonstrating its resilience to market volatility. The maximum drawdown is also lower at 11.87%, indicating that the model better withstands market dips in practical use. Notably, the Sharpe ratio is elevated at 10.58, further underscoring the model's strong performance on a risk-adjusted basis in real-world conditions. When comparing the actual case to the equal-weighted benchmark, our optimized model achieves not only a higher return rate (116.27% versus 65.10%) but also exhibits enhanced stability and risk efficiency. While the equal-weighted portfolio has a lower maximum drawdown (7.22%), it has a higher standard deviation of 22.44%, suggesting more pronounced volatility. In contrast, the actual model, with its comparatively lower standard deviation and significantly higher Sharpe ratio, demonstrates a more favorable risk-return profile, delivering superior returns with a more controlled level of risk. In conclusion, the analysis of these cases demonstrates the strengths of our optimization model in different scenarios. The theoretical case showcases the model's potential under ideal conditions, achieving very high returns albeit with increased volatility. Transitioning to the actual case, we observe that our model continues to perform well in real-world conditions, achieving a substantial return rate of 116.27% while exhibiting greater stability compared to the theoretical scenario. When compared to the equal-weighted case benchmark, our optimized portfolio not only delivers higher returns but also maintains a balanced approach to risk. Overall, this comparison highlights that our model successfully balances growth and stability, making it a robust and effective strategy for portfolio management. The model's ability to achieve superior performance in both ideal and actual contexts underscores its value in real-world applications.

V. Conclusion

We have developed a novel portfolio optimization framework that leverages a multi-factor model for asset returns and employs semi-definite programming (SDP) to maximize wealth while ensuring non-negative returns. By integrating economic factors, specifically the stock index and market interest rates, with asset returns, we have proposed an approach that dynamically adjusts investment allocations over time to optimize the portfolio's performance. Through comprehensive simulations, we demonstrated the effectiveness of the model under both theoretical and real-world scenarios. The theoretical case, based on perfect foresight, achieved the highest returns but also exhibited significant volatility. In contrast, when applied to real-world stock data from 2023, the model produced strong returns with improved stability, as evidenced by a higher Sharpe ratio and lower maximum drawdown compared to the equal-weighted benchmark portfolio. The results highlight that the optimization model outperforms a simple equal-weighted portfolio, achieving superior returns while maintaining a more favorable risk-return profile. The model demonstrates its robustness by successfully balancing growth and stability, even in the presence of market uncertainties. Furthermore, the application of semi-definite programming ensures that the portfolio allocations respect the constraints on risk, making the solution both practical and reliable. Overall, this study not only contributes a powerful optimization tool for portfolio management but also showcases its real-world applicability. By combining sophisticated modeling techniques with practical data, the proposed framework provides an adaptive, scalable solution for managing investments. Future work could explore further refinements, such as incorporating more factors, improving the model's robustness to market shocks, or extending the approach to handle larger, more complex portfolios.

Acknowledgment

Qing-Guo Wang acknowledges the financial support of National Natural Science Foundation of China under Grant 62373060, BNU Talent seed fund, and the Guangdong Provincial Key Laboratory IRADS for Data Science (2022B1212010006).

References

[1] H. Markowitz. (1952). Portfolio selection. *The Journal of Finance*, 7(1), 77–91.

[2] E. F. Fama. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383–417.

[3] F. Black, & R. Litterman. (1992). Global portfolio optimization. *Financial Analysts Journal*, 48(5), 28–43.

[4] R. J. Shiller. (1981). Do stock prices move too much to be justified by subsequent changes in dividends?. *The American Economic Review*, 71(3), 421–436.

[5] J. Y. Campbell, & R. J. Shiller. (2000). Valuation ratios and the long-run stock market outlook. *The Journal of Portfolio Management*, 26(4), 11–26.

[6] R. C. Merton. (1992). *Continuous-time finance*. Blackwell.

[7] W. F. Sharpe. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3), 425–442.

[8] E. F. Fama, “The behavior of stock-market prices,” *The Journal of Business*, vol. 38, no. 1, pp. 34–105, 1965.
[9] A. W. Lo, “The adaptive markets hypothesis: Market efficiency from an evolutionary perspective,” *The Journal of Portfolio Management*, vol. 30, no. 5, pp. 15–29, 2004.
[10] B. G. Malkiel, *A Random Walk Down Wall Street: The Time-tested Strategy for Successful Investing*. WW Norton & Company, 2019.
[11] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[12] J. He, Q.-G. Wang, P. Cheng, J. Chen, and Y. Sun, ”Multi-Period Mean-Variance Portfolio Optimization With High-Order Coupled Asset Dynamics,” *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3106-3119, Aug. 2019.
[13] J. Smith, ”The relationship between stock index changes and stock returns,” *Journal of Financial Economics*, vol. 10, no. 2, pp. 123-145, 2018.
[14] A. Johnson, ”Interest rates and stock market returns,” *International Journal of Finance*, vol. 15, no. 3, pp. 201-220, 2019.
[15] L. Wang, ”Multi-factor models in financial markets,” *Finance Research Letters*, vol. 25, no. 1, pp. 50-65, 2020.
[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, 2013.
[17] G. Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press, 1993.
[18] Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, 2012.
[19] G. Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press, 1993.
[20] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
[21] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Conditions*, M.Sc. Dissertation, Dept. of Mathematics, Univ. of Chicago, 1939.
[22] Vandenberghe, Lieven and Boyd, Stephen, *Semidefinite programming*, SIAM review, 38(1):49–95, 1996.
[23] Güler, O., *Barrier functions in interior point methods*, Mathematics of Operations Research, 21, 860–885, 1996.
[24] E. D. Andersen, K. D. Andersen, *The MOSEK Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm*, in High Performance Optimization, Springer, pp. 197-232, 2000.
[25] Stephen J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, 1997.
[26] Nesterov, Y., Todd, M., *Self-scaled barriers and interior-point methods for convex programming*, Mathematics of Operations Research, 22, 1–42, 1997.
[27] Nesterov, Y., Nemirovskii, A., *Interior-point polynomial methods in convex programming*, SIAM, Philadelphia, 1994.
[28] Karmarkar, N., *A new polynomial time algorithm for linear programming*, Combinatorica, 4(4), 373–395, 1984.
[29] S. A. Ross, *The Arbitrage Theory of Capital Asset Pricing*, *Journal of Economic Theory*, vol. 13, no. 3, pp. 341–360, 1976.
[30] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004. Available: https://web.stanford.edu/~boyd/cvxbook/
[31] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2006.