A Heuristic Approach to Portfolio Optimization with Transaction Costs and Lots

Area of Research : Optimization in Finance

Project - Part II

Submitted By :
Altaf Ahmad
Roll No : 18MA20005

Supervisor:
Dr. Geetanjali Panda,
Department of Mathematics

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
KHARAGPUR-721302

May 2023

Contents

1. Self-Declaration

I hereby declare that the research work presented in this project titled "A Heuristic Approach to Portfolio Optimization with Transaction Costs and Lots" is a genuine and original work carried out by me during the period from January 2023 to May 2023 at the Department of Mathematics, Indian Institute of Technology - Kharagpur. This work is being submitted as a partial fulfillment for the award of the Master of Science in Mathematics and Computing. I have worked under the guidance of Dr. Geetanjali Panda, who provided valuable insights and guidance throughout the project. The matter presented in this project report is original and has not been submitted elsewhere for the award of any other degree to the best of my knowledge. I hereby declare that the work presented in this project report is my own and all sources used have been duly acknowledged.

Altaf Ahmad, 18MA20005
Department of Mathematics
Indian Institute of Technology Kharagpur
May, 2023

2. Acknowledgement

It is with immense pleasure and thankfulness that I extend my heartfelt thanks to everyone who has contributed to the success of this project.

I would like to begin by expressing my deepest gratitude to my advisor, Dr. Geetanjali Panda. Her extensive knowledge and expertise in mathematics, along with her rational style of thinking, have been a constant source of guidance and inspiration for me throughout this project. She has not only taught me how to conduct research with diligence and attention but also encouraged me to approach research with a curious mind.

I would also like to extend my gratitude to the entire math faculty and personnel at the Indian Institute of Technology in Kharagpur for their guidance, assistance, and training. Their support and encouragement have been instrumental in shaping my understanding of this subject and have been invaluable in making this dissertation a success.

I am also indebted to my classmates and friends for their unwavering support and encouragement during the challenging times. Their constant motivation and support have kept me going even when things seemed tough.

Last but not least, I would like to extend my heartfelt gratitude to my parents and brother. Their unwavering love, endless encouragement, and unending support have been the driving force behind my success. Without them, I would not have been able to complete this project.

In conclusion, I am truly grateful to everyone who has contributed to the success of this project. Your support, guidance, and encouragement have made this journey an incredible learning experience, and I am honored to have had the opportunity to work with such wonderful people.

3. Abstract

This project aims to address the challenges of portfolio optimization problems with transaction costs and lots using multi-objective heuristic algorithms. While the Markowitz mean-variance portfolio optimization theory is widely used, it is known to be sensitive to even small changes in data. To overcome this limitation, the proposed model incorporates rebalancing and repair processes into a mean-variance model for portfolio rebalancing optimization problems. The resulting problem is modeled as a non-smooth nonlinear integer programming problem. To solve this problem, a genetic algorithm based on real-value genetic operators is designed. The efficiency of the proposed algorithm is demonstrated through a numerical example. The research project aims to contribute to developing effective portfolio optimization strategies that account for real-world constraints such as transaction costs and minimum transaction lots.

4. Introduction

The mean-variance optimization problem is one of the leading investment theories in finance. Developed by Harry Markowitz in 1952 [13], it is one of the most important contributions to modern portfolio theory. It provides a quantitative framework for constructing an efficient portfolio of assets by considering both the expected return and the risk associated with each asset.

The problem involves selecting a set of assets from a larger set of available assets and determining the optimal allocation of funds to each asset in the selected set. The objective is to maximize the expected return of the portfolio, subject to a constraint on the overall risk of the portfolio.

The risk of a portfolio is measured by the variance of its returns. Markowitz recognized that by diversifying investments across multiple assets, investors could reduce the overall risk of their portfolio without sacrificing expected return.

Markowitz's work revolutionized the way investors approach portfolio management. Prior to his work, investors typically chose assets based on their expected return alone, without considering the risk associated with those assets. Markowitz's approach provided a rigorous methodology for balancing risk and return, which has since become a cornerstone of modern finance.

Since the development of mean-variance optimization, a vast body of research has been conducted on portfolio theory and asset allocation. Researchers have explored variations on Markowitz's original formulation, as well as extensions to consider other factors such as liquidity, transaction costs, and behavioral biases.

This project aims to extend the classic Markowitz mean-variance optimization framework in one such direction by incorporating transaction costs, and lots and the effect of periodic portfolio rebalancing. While the original Markowitz approach provided a powerful framework for constructing an efficient portfolio, it did not consider the costs associated with executing trades, nor did it account for the need to periodically adjust the portfolio to maintain its desired asset allocation.

The inclusion of transaction costs and the consideration of periodic rebalancing is particularly relevant in today's financial landscape, where trading costs are an important consideration and markets are subject to greater volatility and uncertainty.

The report will first provide a brief overview of the Markowitz mean-variance optimization framework, highlighting its strengths and limitations. It will then present the proposed extension, discussing the impact of transaction costs on portfolio optimization and the importance of rebalancing in maintaining the desired asset allocation.

The paper will also provide empirical evidence to support the efficacy of the proposed extension, drawing on historical market data to illustrate the impact of transaction costs and the benefits of periodic rebalancing.

5. Literature review

One of the most difficult issues in the finance industry is portfolio optimization. This challenge becomes even more important when choosing the weights of the portfolio to invest in each asset in order to achieve the risk and return expectations. Each investor may have different preferences for risk-margin ratios, but a reasonable person will always aim to maximize risk-return tradeoffs. Researchers like Harry Markowitz [13][14] have compiled theorized the maximization of returns in a given risk in greater detail. He emphasized the significance of building portfolios to maximize the expected return on uncertainty in portfolio theory. Although this theory has received a lot of praise, some studies [15] have cast doubt on its applicability. Some techniques, including constrained optimization (CO), quadratic programming (QP), linear one programming (LP), and second-order cone programming (SOCP), have been developed and used[7] to address the drawbacks of the Markowitz mean-variance model.

Due to regime change, variance along with the mean, which presupposes normality of the asset returns, typically does not hold in real market circumstances [8]. As a result, financial academics are drawn to investigating novel approaches to calculating risk, particularly the use of downside risk metrics that only include unfavorable circumstances. For instance, given its formulation, semi-variance [19] only takes into account downward or negative departures from the mean. The objective of the downside risk measure is to increase the likelihood that the return on the portfolio will be higher than an acceptable level. Although measures of downside risk are more susceptible to estimation error because they only use a portion of the data [9], they are more accurate than the variance when the return distribution is asymmetric. Financial institutions have recently given VaR and CVaR—two very effective downside risk measures—consideration. VaR calculates the portfolio's loss so that it will only surpass its value with a specific probability (standard probability values are 0.05 or 0.01). The conditional expectation of losses exceeding the VaR value is estimated by CVaR.

The most widely utilized practical restraint that prevents excessive investment in one or more stocks, industries, or sectors is quantity restrictions on the portfolio. The majority of the time, a portfolio manager seeks to limit the total number of assets (cardinality) in the portfolio due to the market's rising size. The cardinality of the portfolio is constrained within integer boundaries in some studies, e.g. [12] [16], whereas in others [5][3] it is confined to a constant number. In an optimization problem, the cardinality constraint mimics investor preferences, and its bounds are typically exogenous variables in the optimization model. Cardinality can, however, also be regarded as a model endogenous parameter. The cardinality constraint simulates investor preferences in an optimization issue, and its bounds are often exogenous variables in the optimization model. However, cardinality can also be thought of as an endogenous parameter in the model. For instance, Anagnostopoulos and Mamanis [2] evaluated the cardinality of the portfolio as the third target in addition to the mean and variance objectives. Another significant realistic restriction for incorporating investors' subjective preferences in the model is the pre-assignment constraint [12], which mandates that specific pre-decided assets always be included in the portfolio.

When it comes to realistic portfolio optimization and rebalancing, transaction cost is a crucial factor. Yoshimoto [21] noted that portfolios can become inefficient when costs are disregarded. The portfolio manager has a hurdle when costs are included in the portfolio optimization model. In the separable cost model, which Lobo et al. [17] investigated, the transaction costs related to security are represented as the sum of fixed and variable costs. In real-world circumstances, fixed costs are employed to simulate brokerage commissions since they are unaffected by trading volume, whereas variable costs are related to changes in asset positions. The sum of each asset's individual costs represents the overall expenses incurred as a result of changes in the positions of all the assets in the portfolio. The separable transaction costs model is the name given to such a model.

The use of evolutionary algorithms (EAs) as alternative methods to tackle portfolio optimization problems under actual constraints is very common. Many EAs were modified in prior studies to handle the mean-variance portfolio optimization problem with real-world constraints. Heuristics approaches based on genetic algorithms, tabu searches, and simulated annealing were proposed by Chang et al. [4]. By adding a subset optimization phase, Woodside-Oriakhi et al. [20] provided an enhancement over these techniques. For the cardinality-limited portfolio optimization using Particle Swarm Optimization (PSO), Cura [6] developed a heuristic technique. With VaR or CVaR as risk measures and realistic constraints, evolutionary algorithms are also used to optimize risk-return portfolios. The use of PSO and Genetic Algorithm (GA) for portfolio optimization with VaR aim was demonstrated by Dallagnol et al. [22]. The Fireworks algorithm for mean-VaR and mean-CVaR portfolio models was recently introduced by Zhang and Liu [18]. Krink and Paterlini [11] investigated three bi-objective portfolio optimization models with variance, VaR, or projected shortfall as the risk metric in the framework of Multi-Objective Evolutionary algorithms. The bi-objective mean-risk problem in the presence of quantity, cardinality, and class restrictions was also studied by Anagnostopoulos and Mamanis [1]. In the study, three cutting-edge MOEAs—NSGA-II, Pareto Envelope-based Selection Algorithm (PESA), and Strength Pareto Evolutionary Algorithm 2 (SPEA2)—were evaluated using several risk measures, including variance, VaR, and predicted shortfall.

So, as we have seen, optimization models with various constraints and objectives have been used to represent portfolio selection challenges. The most in-depth research focuses on mean-variance models. In several investigations, Mean-VaR and Mean-CVaR models were also taken into account. These models were also changed by adding plausible restrictions that occur in the financial sector. A number of the models included bound and cardinality restrictions in addition to the apparent budget constraint. Most studies used a narrow strategy for portfolio optimization without taking the rebalancing issue into account. Some studies also employed a multi-period approach, monitoring risk-return dynamics and rebalancing the portfolio on a regular basis or whenever a pre-determined event occurs.

However, the cost of rebalancing significantly reduces portfolio returns, particularly in a multi-period setting. As a result, taking into account transaction costs is obvious when choosing an investment strategy. In general, non-linear functions of changes in investments characterize transaction cost models. Because of this, traditional optimization methodologies are inapplicable when realistic restrictions and a transaction cost control mechanism are included non the optimization model. It is necessary to build specialized algorithms that are especially suited to the task at hand. The most well-liked and desirable candidate for these improvements is MOEAs. In recent studies, appealing MOEA frameworks have been modified to address the cardinality-constrained portfolio optimization challenge.

6. Preliminaries

The Markowitz mean-variance optimization framework is a mathematical approach to constructing a portfolio of assets that seek to maximize expected returns while minimizing risk. The Markowitz framework requires that an investor know the expected returns and risks of each asset in the portfolio as well as the correlations between the assets. With this information, the investor can construct an efficient frontier, which is a curve that represents the optimal portfolios that offer the highest expected return for a given level of risk or the lowest risk for a given level of expected return.

6.1 Notations

- Consider Assets : $A_0, A_1, A_2, ..., A_n$
- Suppose X is the total principal amount distributed amongst the assets at a time t_0
- $\sum_{i=1}^{n} X_i = X$. If $X_i < 0$, then it is called short selling.
- Consider $w_i = X_i/X$ as the ratio of investment in asset i. Thus, $\sum_{i=1}^{n} w_i = 1$ or $e^T w = 1$
- r_i is the rate of return of the asset i.
- $\mu_i = E(r_i) =$ Expected return of asset i for the time T $= 1/(n-1) \times \sum_{i=1}^{n} (r_{i,j} - r_{i,j-1})/(r_{i,j-1})$
- The rate of return of the portfolio = R = $\sum_{i=1}^{n} w_i r_i$
- Total Expected Return of the portfolio P = $\mu_P = \sum_{i=1}^{n} w_i \mu_i = w^T \mu$

6.2 Risks of Portfolio

There are several types of risk functions used in portfolio analysis. Some of the most commonly used risk functions are:

1. Variance Risk: The variance of a portfolio measures how much the actual returns of the portfolio are likely to deviate from the expected returns. A portfolio with a high variance is considered to be riskier than one with a low variance. This gives a quadratic function.
2. Mean Absolute deviation: Measures the average absolute deviation of the actual returns from the expected returns of a portfolio. Unlike variance and standard deviation, MAD gives equal weight to positive and negative deviations from the expected return, making it a useful measure of risk for investors who are more concerned about the downside risk of a portfolio. This gives a linear function.

3. Min-Max Risk: It is a risk measure used in portfolio optimization that aims to minimize the worst-case scenario for a portfolio. This measure seeks to minimize the maximum possible loss that a portfolio could experience over a specified time horizon with a given level of confidence. This can be converted into a linear function.

4. Value at Risk: VaR is a measure of the maximum loss that a portfolio is expected to suffer with a certain level of probability over a specified time horizon. It provides an estimate of the worst-case scenario for a portfolio.

5. Conditional value-at-risk (CVaR): CVaR is similar to VaR, but it measures the expected loss beyond the VaR level. It provides an estimate of the average loss that a portfolio is expected to suffer in the worst-case scenario.

6.3 Variance Risk

In this framework, the measure of risk is the variance of returns. The variance of the portfolio can be denoted as

$$\sigma^2 = E[(R - E(R))^2]$$

For asset i, variance $= \sigma_i^2 = E[(r_i - E(r_i))^2]$

And covariance of the return of asset i and asset $j = \sigma_{ij} = E[(r_i - E(r_i))(r_j - E(r_j))]$. So, for the portfolio, we have

$$\sigma^2 = E[(R - E(R))^2] = E\big(\sum_{i=1}^{n} w_i r_i - \sum_{i=1}^{n} w_i \mu_i\big)^2$$

$$= E(\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j (r_i - \mu_i)(r_j - \mu_j)) = (\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j E[(r_i - \mu_i)(r_j - \mu_j)])$$

$$= (\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}) = w^T \Omega w$$

Where, $w = (w_1, w_2, ..., w_n)^T$,

$$\Omega = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & . & . & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & . & . & \sigma_{2n} \\ . & . & . & . & . \\ \sigma_{n1} & \sigma_{n2} & . & . & \sigma_n^2 \end{pmatrix}$$

So, for a given expected rate of return R, we can write the framework as

$$\text{Minimize}: \ w^T \Omega w$$

$$\text{Subject to}: \ \mu^T w = R, e^T w = 1$$

6.4 Limitations

Professionals generally agree that the Markowitz mean-variance method works. It offers the idea of risk and return in a straightforward manner and is simple to understand. This strategy hasn't yet been shown to be effective outside of samples, though. There are some limitations to mean-variance optimization that we want to highlight here :

- Sensitivity to input assumptions: Markowitz's mean-variance model is sensitive to the accuracy of the input assumptions used to estimate the expected returns, volatilities, and correlations of the assets in the portfolio. Small errors in these inputs can lead to significant changes in the optimized portfolio and can result in suboptimal allocations.
- Focus on volatility: The mean-variance model focuses primarily on minimizing portfolio volatility, but this may not always align with an investor's actual preferences or risk tolerance. For example, investors with a longer investment horizon may be willing to tolerate higher volatility in exchange for potentially higher long-term returns.
- Assumption of the normal distribution: The mean-variance model assumes that asset returns are normally distributed, but this is often not the case in practice. Asset returns may be skewed, have heavy tails, or exhibit other forms of non-normal behavior, which can impact the validity of the model's output.
- Neglect of tail risk: The mean-variance model does not explicitly consider the tail risk or the risk of extreme events that may have a significant impact on portfolio returns. In some cases, minimizing tail risk may be a more important objective than minimizing overall portfolio volatility.
- Failure to account for transaction costs: The mean-variance model does not explicitly incorporate transaction costs, which can have a significant impact on portfolio performance. Ignoring transaction costs can lead to sub-optimal portfolio allocations.
- The "best" assets make up a small percentage of the assets in MV-optimized portfolios. Due to the over-weighting of assets with high predicted returns, optimization's intended benefit of diversity is lost.

6.5 Genetic Algorithms

Holland [10] described genetic algorithms as processes, based on the mechanisms of genetic mutation of DNA, as they occur in nature and describe their applicability for a number of different situations. A population in the context of portfolio optimization is made up of numerous different portfolios, or potential solutions. Fitness describes the value of each portfolio's objective function but may also include other metrics. One iteration of the updating cycle, which entails selection, mutation, and cross-over operations, is referred to as a generation. Which historical population portfolios are used to create new candidates for the population of the following generations depends on selection. Based on their assessments, this is possible since portfolios that are fitter have a higher likelihood of reproducing or passing directly to the next generation. The update stage's cross-over operation then decides how the chosen parent portfolios are combined to create a new child portfolio, or how to build new, potentially better solutions on top of the existing ones. Last but not least, the process of mutation includes unpredictability and randomness which is introduced externally.

The algorithm manipulates a population with a constant size in an iterative search for the best answer. Chromosomes are candidate points in this population. The chromosomes begin to compete with one another as a result of this process. Each chromosome encodes a potential answer to the problem that has to be solved; it is made up of a group of components called genes that can have different values. A fresh population of the same size is produced at each iteration (generation). The better chromosomes that "adapted" to their environment as shown by the selection function make up this generation. The chromosomes will gradually gravitate in the direction of the optimal selective function. The conception of the new population is made by applying the genetic operators which are selection, crossover, and mutation.

- **Selection**: The new individuals are chosen as follows: Determine each individual's likelihood of reproducing as

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i}$$

where f_i is the fitness of the individual i. This can be calculated using some fitness functions. Here n is the size of the population. Each time a single chromosome or element from the population is selected for a crossover with another selected element for the creation of the progeny. T This is done a total of n times with the elements randomly selected from the parent generation so that now there are a total of $2n$ elements in the population. Each time a single chromosome or element from the population is selected for a crossover with another selected element for the creation of the progeny. This can be calculated using some fitness functions. The choice of crossover operator depends on the data structure used to store genetic information and the problem being solved. Different algorithms in evolutionary computation may use different data structures to store genetic information, and each genetic representation can be recombined with different crossover operators.

- **Crossover**: It is a genetic operator that is used to combine the genetic information of two parents to generate new offspring. This operator follows a specific process where the population resulting from selection is divided into two parts, and each pair formed will undergo the crossover with a certain probability P_c. The crossover operator is one way to stochastically generate new solutions from an existing population, and it is analogous to the reproduction and biological crossover. There are many different types of crossover operators that exist in the literature, including single-point crossover, two-point crossover, and arithmetic crossover. The choice of crossover operator depends on the data structure used to store genetic information and the problem being solved. For example, traditional genetic algorithms store genetic information in a chromosome represented by a bit array, and crossover methods for bit arrays are popular. Two-point crossover is a specific case of an N-point crossover technique, where two random points are chosen on the individual chromosomes, and the genetic material is exchanged at these points. Another type of crossover is a uniform crossover, where each gene (bit) is selected randomly from one of the corresponding genes of the parent chromosomes.

  The crossover operator is a significant phase in a genetic algorithm, and it is used to avoid the exact duplication of the parents from the old population in the new offspring. This ensures that the new population is diverse and can explore a larger search space. The choice of the crossover probability P_c is an important parameter in a genetic algorithm, and it should be set based on the problem being solved. A high crossover probability can lead to premature convergence, while a low crossover probability can lead to slow convergence. Therefore, the crossover probability should be set based on the problem being solved and the characteristics of the population.

- **Mutation**: It is a genetic operator used to maintain the genetic diversity of the chromosomes of a population of a genetic or, more generally, an evolutionary algorithm. The mutation operator helps protect against the problem of premature convergence by maintaining diversity in the population. The mutation operator randomly changes some bits, with a certain probability P_m, to produce a new offspring. This creates new adaptive solutions and helps avoid local optima. For example, in binary encoding, one or more randomly chosen bits can be switched. The choice of mutation probability P_m is an important parameter in a genetic algorithm, and it should be set based on the problem being solved. A high mutation probability can lead to a loss of good solutions, while a low mutation probability can lead to slow convergence. Therefore, the mutation probability should be set based on the problem being solved and the characteristics of the population.

They are a type of metaheuristic algorithm that is commonly used to generate high-quality solutions to optimization and search problems. They are more flexible than most search methods because they require only information concerning the quality of the solution produced by each parameter set (objective function values). This differs from many optimization methods which require derivative information, or even more, complete knowledge of the problem structure and parameters. In a genetic algorithm, a population of candidate solutions to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered. A standard representation of each candidate solution is as an array of bits (also called bit set or bit string).

The main benefit of genetic algorithms is that you don't have to predetermine every aspect of an issue. In fact, one doesn't even need to be familiar with all the specifics. A fitness function that represents the issue we're trying to address is used to assess potential solutions. Then, in order to generate fresh candidate solutions, we define an evolution procedure. Combining good solutions (solutions that rank highly on the fitness scale), according to the theory, should produce even better answers. We aim to identify better solutions by modifying the candidate and adding some noise. The process of evolution includes selecting the individuals who will make up the following generation of solutions.

When using genetic algorithms, there are two main steps.

1. When writing your problem, make it as simple as possible for answers to be coded and altered automatically. Evolution's "mutate and reproduce" phase is represented by this.

2. Establish a fitness function, f. This function will be used to compare and order the solutions. The "survival of the fittest" aspect of evolution is at play here.

After the initial population is created, the genetic algorithm proceeds through a series of steps to generate new candidate solutions. The algorithm can be described in a high level as follows:

Algorithm 1 Pseudo Code

```
Draw a random population of n candidates. This is the old population.
Generate an intermediate population of n candidates.
Compare and rank the 2n candidates.
Choose the n highest scoring candidates as the new population.
Go to step 2.
```

6.5.1 Defining the Problem

In order to utilize a genetic algorithm for the purpose of finding optimal portfolios of assets, it is essential to formulate the problem in a manner that is amenable to such an approach. The objective is to identify portfolios that exhibit strong performance as measured by our fitness criteria.

To accomplish this, we must first define the parameters that will be used to represent the portfolio, such as the individual assets included and their respective weights. We then establish the fitness function that will evaluate the quality of a given portfolio based on its performance with respect to our desired criteria. Our fitness function could be based on the Sharpe ratio, which is a commonly used metric for evaluating the risk-adjusted return of a portfolio. The Sharpe ratio is calculated by dividing the portfolio's excess return (i.e., its return above the risk-free rate) by its standard deviation. A higher Sharpe ratio indicates better performance. We could then use this value as the fitness score for our genetic algorithm. The goal of the algorithm would be to maximize the Sharpe ratio by evolving the weights of the stocks in the portfolio over successive generations.

Next, we create an initial population of portfolios, which serves as the starting point for our genetic algorithm, and the portfolios in the population are used to generate new portfolios in each generation. In our case, we are going to use a random number generator to create a set of portfolios that satisfy the constraints. For each portfolio, randomly assign weights to the assets in the universe such that the sum of the weights equals 1 (i.e., the portfolio is fully invested). Through successive generations of the algorithm, the portfolios evolve and improve in accordance with the fitness function, ultimately converging toward the optimal portfolio(s). The convergence of the portfolios to the optimal solution(s) can be influenced by several factors, including the size of the population, the number of generations, and the selection pressure. Larger population size and a longer running time can provide more opportunities for the algorithm to explore the search space and improve the portfolios. Higher selection pressure can also accelerate convergence by favoring the fittest solutions in each generation.

7. Experiment, Methodology, and Results

The Mean-Variance (MV) method, as typically used, is a static process where a single portfolio is selected based on past data that is optimal for the investor's mean/variance preferences at a particular point in time. However, to achieve dynamic portfolio optimization, we need to continuously re-evaluate and adjust the portfolio based on changes in market conditions and the investor's preferences.

In this project, our goal is to develop dynamic portfolio optimization methods that allow us to choose a portfolio initially and then regularly rebalance it to ensure it remains optimal according to our fitness standards. Rebalancing involves periodically adjusting the asset allocation to maintain the desired risk and return characteristics of the portfolio.

While we could simply use MV optimization over time, it has been shown to be ineffective due to high reallocation costs. As such, we aim to develop and compare different dynamic strategies against standard techniques, some of which are simple yet surprisingly effective.

By incorporating rebalancing into the portfolio optimization process, we can ensure that the portfolio remains aligned with our investment objectives and adapts to changes in market conditions over time. This requires a dynamic approach that considers current market conditions, as well as the investor's changing risk tolerance and investment goals. Effective dynamic portfolio optimization methods will help investors achieve their long-term investment objectives while minimizing costs and mitigating risk.

7.1 Data and Framework

For this project, we have selected a dataset consisting of 30 stocks from the National Stock Exchange (NSE) in India. These stocks were chosen based on their liquidity and popularity among investors. We collected data for a time period of 100 days, starting on April 1, 2022, and ending on August 26, 2022.

The dataset includes a variety of stock types, including large-cap, mid-cap, and small-cap stocks from different industries such as technology, healthcare, and finance. The use of real-world data from the NSE will provide insights into the performance of different investment strategies and allow for a more realistic evaluation of the effectiveness of the methods developed in this project. To ensure the quality of our data, we used reliable sources such as NSE and official government statistics. We also checked the data for any inconsistencies or outliers and corrected them as necessary. The use of real-world data from the NSE will provide insights into the performance of different investment strategies and allow for a more realistic evaluation of the effectiveness of the methods developed in this project.

7.2 Articulation of the Problem

Let S be the set of securities in which we aim to invest an initial capital of $C_0 \leq C \leq C_1$, where $s = |S|$ is the total number of securities. A portfolio can be represented by a vector $k = (k_1, k_2, ..., k_s)$,

Date	526721	541729	524667	513430	500285	500112	519156	523618	532725	...	517354	532454	500870	500875	507155	502168	500463
1-April-2022	63.9	2178.10	1061.20	69.00	55.20	492.30	1420.35	317.10	2881.40	...	1153.95	757.15	101.90	249.55	61.20	178.00	772.90
4-April-2022	69.8	2293.30	1110.40	63.35	57.50	510.00	1459.20	323.80	2911.00	...	1188.20	758.60	104.50	253.40	62.50	193.00	772.90
5-April-2022	67.0	2374.00	1128.35	60.20	57.50	513.50	1535.00	332.30	2876.05	...	1180.00	771.00	104.70	255.80	64.50	199.90	783.00
6-April-2022	67.0	2330.00	1135.20	57.20	59.00	507.05	1475.00	337.00	2997.00	...	1258.00	765.00	104.35	259.00	65.30	206.00	791.05
7-April-2022	69.9	2306.10	1164.85	54.35	58.60	511.80	1490.00	347.00	2999.00	...	1251.70	772.00	104.05	259.00	64.80	207.90	774.60
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22-August-2022	87.8	2132.05	1519.95	38.10	46.00	518.10	2289.90	276.30	3372.60	...	1345.00	729.00	116.55	309.05	62.95	177.85	129.10
23-August-2022	81.4	2101.75	1527.95	37.50	45.60	506.20	2337.60	277.95	3420.00	...	1270.00	719.15	114.00	313.80	59.05	177.90	125.00
24-August-2022	88.5	2140.00	1633.70	35.30	46.70	519.50	2506.00	279.60	3418.00	...	1291.00	728.00	113.00	315.80	62.00	179.85	128.25
25-August-2022	89.0	2113.40	1677.95	38.25	47.55	521.50	2700.00	281.55	3440.00	...	1309.05	743.50	114.60	315.05	61.70	180.35	132.05
26-August-2022	86.0	2111.75	1629.95	34.65	46.50	523.05	2550.90	284.55	3413.40	...	1324.90	743.00	114.95	313.90	63.05	182.10	134.00

Figure 7.1: Dataset stocks

where k_j denotes the number of shares of security j. Let r_j represent the random rate of return on security $j \in S$, n_j^s be the minimum transaction lot when selling security j, and n_j^b be the minimum transaction lot when buying security j. Let $R_j = E(r_j)$ denote the expected rate of return on security j, and let $\sigma_{ij} = cov(r_i, r_j)$ be the covariance between the returns of security i and security j.

Furthermore, let u_j represent the maximum amount of capital that can be invested in security j, d_j represent the fixed proportion parameter of transaction cost associated with security j, and p_j represent the quoted price of security j. We can define $y_j = (p_j k_j)/C$ as the proportion of the portfolio invested in security j.

The expected return $R(k)$ and the variance $V(k)$ can be calculated as follows:

$$R(k) = (\sum_{j \in S} R_j p_j k_j - \sum_{j \in S} d_j p_j |k_j - k_j^0|)/C$$

$$V(k) = \sum_{i \in S} \sum_{j \in S} \sigma_{ij} y_i y_j$$

Our goal is to maximize the expected return while minimizing the variance of the portfolio. This is known as the mean-variance portfolio optimization problem. However, we also need to consider the constraints imposed by transaction costs, minimum and maximum investment limits, and minimum transaction lots when buying and selling securities. By incorporating these constraints, we can formulate an optimization problem that selects the optimal portfolio that maximizes the expected return while minimizing the variance, subject to the constraints.

The rebalancing problem can be stated as follows:

$$\text{minimize } f(k) = -\lambda . R(k) + (1 - \lambda) . \omega . V(k)$$

$$\text{s.t.} \quad C_0 \leq \sum_{j \in S} p_j k_j = C \leq C_1$$

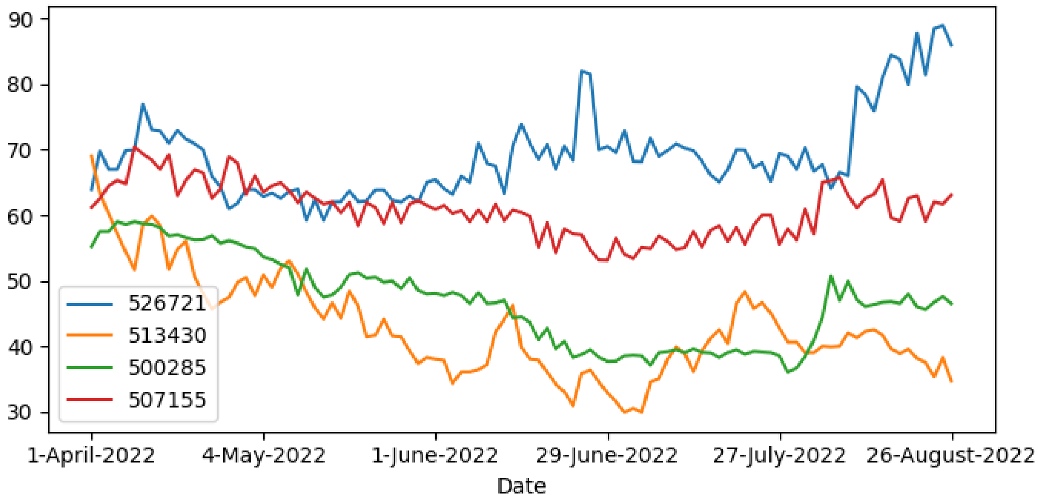

Figure 7.2: Example stock variation

$$
\begin{aligned}
0 \le p_j k_j \le u_j, \quad & \text{for all } j \in S \\
n_j^b | (k_j - k_j^0), \quad & \text{for all } j \in S, \ \text{if } k_j > k_k^0, \\
n_j^s | (k_j^0 - k_j^{)}, \quad & \text{for all } j \in S, \ \text{if } k_j < k_k^0,
\end{aligned}
$$

Here, the parameter λ is varying in [0,1] and ω is a scaling parameter.

7.3 Algorithm Formulation

Representation Structure: The way in which portfolios are represented in the genetic algorithm is by using a structure that consists of a string of non-negative integer numbers. Each number in the string represents a specific security, and its value corresponds to the number of shares of that security in the portfolio. Therefore, the length of the string is equal to the number of securities included in the portfolio.

Repair Process: For an existing portfolio $k^0 = (k_1^0, ..., k_s^0)$,a real vector $x = (x_1, ..., x_s)$ can be repaired into a new portfolio by the following formula:

$$
d_i = \begin{cases} \left[\begin{matrix} \frac{x_i - k_i^0}{n_i^b} \\ \\ \frac{k_i^0 - x_i}{n_i^s} \end{matrix} \right] & \begin{matrix} \text{if } \ x_i \ge k_i^0 \\ \\ \text{if } \ x_i \le k_i^0 \end{matrix} \end{cases}
$$

$$
m_i = \begin{cases} k_i^0 + d_i \cdot n_i^b & \text{if } \quad x_i \ge k_i^0 \\ k_i^0 - d_i \cdot n_i^s & \text{if } \quad x_i \le k_i^0 \end{cases}
$$

$$
m_i' = \begin{cases} k_i^0 + (d_i + 1) \cdot n_i^b & \text{if } \quad x_i \ge k_i^0 \\ k_i^0 - (d_i + 1) \cdot n_i^s & \text{if } \quad x_i \le k_i^0 \end{cases}
$$

$$
k_i = m_i \text{ or } m_i' \text{ randomly.}
$$

This process ensures that the resulting portfolio k satisfies the transaction lot requirements n_j^b and n_j^s for all securities j and remains within the bounds of the investable capital C_0 and C_1.

Algorithm 2 GA using Simulated Binary Crossover (SBX) and Polynomial Mutation (PM) operators

Require: $populationSize, P_c, P_m, gen$

1: Initialize population with $populationSize$ individuals
2: **for** $g \leftarrow 1$ to gen **do**
3: Apply SBX crossover and PM mutation operators to each individual in population
4: Repair the individuals using the repair process
5: Evaluate the fitness of each individual
6: Select individuals using binary tournament selection
7: **end for**

7.4 Results

When we run the Markovitz model on our given dataset of stocks, we are able to obtain an efficient frontier. The efficient frontier is a graphical representation of the set of optimal portfolios that offer the highest expected return for a given level of risk. In other words, the efficient frontier is the curve that marks the boundary of the set of portfolios that are optimal for a given level of risk.

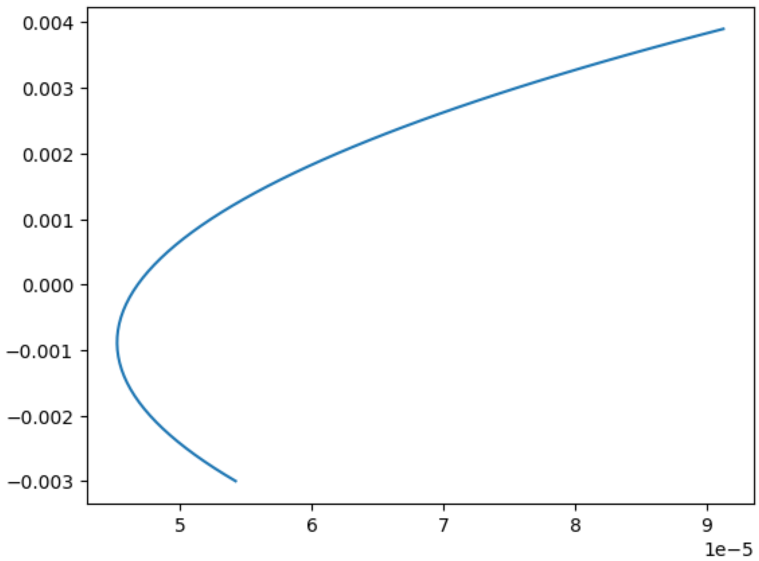

Figure 7.3: Efficient Frontier

Upon running the algorithm on the dataset of stocks from the National Stock Exchange (NSE) with various values of lambda, we obtain a comprehensive set of risk and return values. These values provide insights into the potential risks and expected returns associated with each lambda value. By analyzing the risk-return trade-off for each lambda, we can identify the optimal value of lambda that maximizes returns while minimizing risks.

The algorithm can be refined and tested iteratively to identify the most suitable lambda value that maximizes returns while minimizing risks. This process can involve adjusting the parameters of the algorithm or incorporating additional data sources to enhance its predictive power.

λ	0.0	0.2	0.4	0.6	0.8	1.0
Return	1.1284e-1	1.1356e-1	1.1522e-1	1.1858e-1	1.2000e-1	1.2007e-1
Risk	4.1912e-4	4.1957e-4	4.2757e-4	4.6165e-4	4.9867e-4	5.1007e-4
Fitness	4.1912e-2	1.0854e-2	-2.0434e-2	-5.2681e-2	-8.6028e-2	-1.2007e-1

The graph presented in Figure 7.4 illustrates the convergence of fitness values for a population over a series of epochs. The fitness values represent the effectiveness of each individual in the population at solving a given problem. As the epochs progress, the fitness values of the population begin to converge toward an optimal value.

It is important to note that the rate of convergence can be influenced by various factors such as the size of the population, the selection criteria, and the mutation rate. Therefore, the convergence pattern observed in the graph can provide insights into the effectiveness of the algorithm and help guide further improvements.

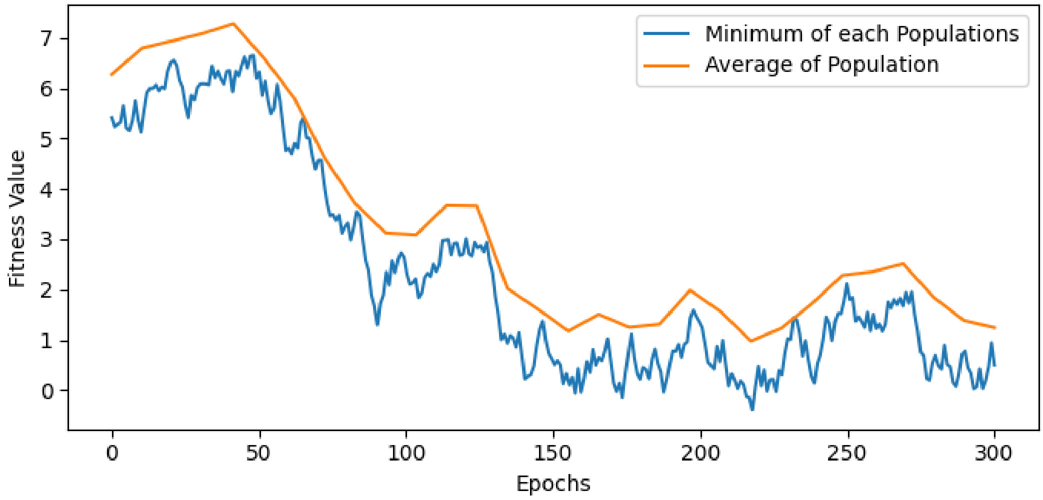

Figure 7.4: Convergence of Fitness Values of Population for $\lambda = 0.4$

```
[ 0.02500582  0.00915283 -0.01831587  0.02109533 -0.04228298  0.10855476
 -0.00063754  0.07376112  0.08724174  0.00566116  0.1366442  -0.03282101
 -0.02861861 -0.03075123  0.03287929  0.00048031  0.08934613  0.03307092
 -0.13800597 -0.04518024 -0.03046644  0.11822033  0.26606891  0.24781095
  0.02324435  0.01374473 -0.0046624   0.01456032  0.06813642 -0.00293733]
```

Figure 7.5: Example of Optimal Portfolio with $\lambda = 0.4$

8. Conclusion and Future Work

The project implements a multi-objective portfolio optimization model that considers risk and returns as objectives while incorporating transaction lots as constraints. To address the limitations of constraint-handling approaches in evolutionary algorithms (EAs) when dealing with problems with equality constraints, a repair algorithm is utilized in this implementation.

The proposed repair algorithm is specifically designed to handle the equality constraint set of the portfolio optimization model, while also considering the general specification of a separable transaction cost model. Together with the candidate generation technique employed in this study, the repair algorithm can effectively handle all the constraints of the optimization model without requiring the use of conventional constraint handling approaches in EAs.

8.1 Improvement and Future Work

This approach was guided by both the existing literature on the topic as well as our own experiences and insights. We recognize that this is a complex problem that requires a multidisciplinary approach, and we are committed to utilizing all available resources and knowledge to find the best possible solutions.

- **Estimation** Historical returns have been used as estimators for future returns, but this method has proven to be unreliable. Other methods include shrinking the vector of expected returns towards an empirical value based on asset managers' beliefs. Also, to estimate the covariance matrix, we have relied solely on naive estimators of V. However, the historical covariance matrix we used as an estimator is known to be highly unstable and unreliable. To address this issue, we aim to shrink V towards a stable and specific matrix. Typically, we can achieve this by projecting the sample V onto a smaller subspace of admissible matrices that are positive definite.

- **Genetic Algorithm** We prioritize convergence over-diversification in our selection process. This involves utilizing the best member of the population to generate new members, which focuses on generating solutions that are similar to the current best member. While this approach may lead to potentially missing out on good solutions located elsewhere in the search space, it is necessary to ensure convergence of the population.

  If we were to use a random member to generate new members instead, we would need to define other criteria for stopping the genetic algorithm. One possible criterion is to track the time elapsed since the last fitness improvement. If this time interval becomes too long, then we terminate the optimization process.

Despite our extensive efforts, we have yet to discover a groundbreaking approach to constructing portfolios. However, we have gained valuable insights into the crucial role of uncertainty in the process. Regardless of the method employed, it is crucial to consider confidence intervals and sensitivity tests to ensure the reliability of the optimization results. It is also important to establish clear limits and constraints for the portfolio optimization methods employed.

Even with provided data and simple functions, we have observed that results can fall short of expectations, and noise can significantly impact the optimization process. Therefore, a whole field has emerged, devoted to filtering out noise and identifying genuine information. This involves the development of models and the application of simplifying assumptions. However, there is always a trade-off between model simplicity and the constraints imposed by real-world data. Shrinking covariance matrices can lead to cleaner results, more stable input parameters, and improved theoretical models. Nonetheless, this can also distance us from the true nature of the data.

Ultimately, practitioners face the challenge of back-testing the optimization method employed. If it yields superior returns, then little else matters. However, it is essential to remain aware of the limitations and potential biases of the chosen approach. The identification of a successful portfolio optimization method requires a careful balance between theoretical assumptions, real-world data constraints, and empirical performance.

Our aim was to develop a more effective approach to portfolio management that made logical investment sense, with each factor serving a clear purpose. We conducted several tests to evaluate the performance of our strategies, only to arrive at a similar conclusion to that of other researchers before us. To our disappointment, we discovered that all of our proposed strategies were outperformed by the simple and straightforward equally weighted approach.

Even over different time periods, we found that no strategy outperformed the market portfolio. This led us to the realization that investing in the entire market would yield superior results compared to our more complex and sophisticated strategies. Moreover, the explanations for the success of the equally weighted approach are much more straightforward and easy to understand, compared to the complicated and nuanced explanations required for our strategies.

While this outcome may seem discouraging, it is essential to recognize the significance of these findings. We must remain open to the possibility that our approach may not always be the best one and that sometimes, the simplest approach can yield the most fruitful results. It is also important to note that our research has contributed to the growing body of knowledge surrounding portfolio management and has helped us to better understand the factors that impact investment outcomes.

Bibliography

[1] Konstantinos Anagnostopoulos and Georgios Mamanis. Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science*, 8(3):259–279, August 2011.

[2] K.P. Anagnostopoulos and Georgios Mamanis. A portfolio optimization model with three objectives and discrete variables. *Computers and Operations Research*, 37:1285–1297, 07 2010.

[3] K.P. Anagnostopoulos and Georgios Mamanis. The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Syst. Appl.*, 38:14208–14217, 05 2011.

[4] T. J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers and Operations Research*, 27(13):1271–1302, 2000.

[5] S.C. Chiam, Abdullah Al-Mamun, and Y.L. Low. A realistic approach to evolutionary multiobjective portfolio optimization. pages 204 – 211, 10 2007.

[6] Tunchan Cura. Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications*, 10:2396–2406, 08 2009.

[7] Marcus Davidsson. Portfolio optimization and linear programming. *ERN: Optimization Techniques; Programming Models; Dynamic Analysis (Topic)*, 2011.

[8] Frank J Fabozzi, Sergio M Focardi, and Petter N Kolm. *Financial Modeling of the Equity Market: From CAPM to Cointegration.* John Wiley & Sons, Hoboken, NJ, 2006.

[9] Henk Grootveld and Winfried Hallerbach. Variance vs downside risk: Is there really that much difference? *European Journal of Operational Research*, 114(2):304–319, April 1999.

[10] John H. Holland. Genetic algorithms. *Scientific American*, 267(1):66–73, 1992.

[11] Thiemo Krink and Sandra Paterlini. Multiobjective optimization using differential evolution for real-world portfolio optimization. *Computational Management Science*, 8:157–179, 04 2009.

[12] Khin Lwin, Rong Qu, and Graham Kendall. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 24:757–772, 2014.

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

[14] Harry M. Markowitz. Foundations of portfolio theory. *The Journal of Finance*, 46(2):469–477, 1991.

[15] Kyong Joo Oh, Tae Yoon Kim, and Sungky Min. Using genetic algorithm to support portfolio optimization for index fund management. *Expert Systems with Applications*, 28(2):371–379, February 2005. Funding Information: This research was financially supported by Hansung University in the year of 2003. We are very grateful to Chun Soo Park and Jung Soon Lee, who work for Hyundai Securities Co., for their careful comments. In addition, we would like to express our thanks to Yang Jin Kim, a devoted librarian of Hansung University.

[16] G. A. Vijayalakshmi Pai and Thierry Michel. Evolutionary optimization of constrained k-means clustered assets for diversification in small portfolios. *IEEE Transactions on Evolutionary Computation*, 13(5):1030–1053, 2009.

[17] Miguel Sousa Lobo, Maryam Fazel, and Stephen Boyd. Portfolio optimization with linear and fixed transaction. *Annals OR*, 152:341–365, 03 2007.

[18] Ying Tan and Yuanchun Zhu. Fireworks algorithm for optimization. In Ying Tan, Yuhui Shi, and Kay Chen Tan, editors, *Advances in Swarm Intelligence*, pages 355–364, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[19] Alexander Thiele, Svetlozar T. Rachev, and Frank J. Fabozzi. *Robust Portfolio Optimization and Management*. John Wiley & Sons, Hoboken, NJ, 2007.

[20] M. Woodside-Oriakhi, C. Lucas, and J.E. Beasley. Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research*, 213(3):538–550, 2011.

[21] Atsushi Yoshimoto. The mean-variance approach to portfolio optimization subject to transaction costs. *Journal of the Operations Research Society of Japan*, 39(1):99–117, 1996.

[22] S. Yourdkhani. Portfolio management by using value at risk (var) (a comparison between particle swarm optimization and genetic algorithms). *Life Science Journal*, 11:15–26, 01 2014.