Springer Optimization and Its Applications 163

Elissaios Sarmas
Panos Xidonas
Haris Doukas

Multicriteria Portfolio Construction with Python

Springer

Springer Optimization and Its Applications

Volume 163

Aims and Scope

Optimization has continued to expand in all directions at an astonishing rate. New algorithmic and theoretical techniques are continually developing and the diffusion into other disciplines is proceeding at a rapid pace, with a spot light on machine learning, artificial intelligence, and quantum computing. Our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in areas not limited to applied mathematics, engineering, medicine, economics, computer science, operations research, and other sciences.

The series **Springer Optimization and Its Applications (SOIA)** aims to publish state-of-the-art expository works (monographs, contributed volumes, textbooks, handbooks) that focus on theory, methods, and applications of optimization. Topics covered include, but are not limited to, nonlinear optimization, combinatorial optimization, continuous optimization, stochastic optimization, Bayesian optimization, optimal control, discrete optimization, multi-objective optimization, and more. New to the series portfolio include Works at the intersection of optimization and machine learning, artificial intelligence, and quantum computing.

Volumes from this series are indexed by Web of Science, zbMATH, Mathematical Reviews, and SCOPUS.

More information about this series at http://www.springer.com/series/7393

Elissaios Sarmas • Panos Xidonas • Haris Doukas

Multicriteria Portfolio Construction with Python

Springer

Elissaios Sarmas
Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece

Panos Xidonas
ESSCA École de Management
Paris, France

Haris Doukas
Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece

ISSN 1931-6828 ISSN 1931-6836 (electronic)
Springer Optimization and Its Applications
ISBN 978-3-030-53742-5 ISBN 978-3-030-53743-2 (eBook)
https://doi.org/10.1007/978-3-030-53743-2

Mathematics Subject Classification: 90-XX, 90-10, 90Bxx, 90Cxx

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The motivation for writing this book came from our years of previous research on multicriteria portfolio management. This book offers a unified methodological framework for portfolio management, integrating multicriteria methods with investment analysis techniques. It results in an outcome that will assist both academics and practitioners, in their everyday routine, since it also includes an extensive implementation of multicriteria decision aid methods in the Python programming language, bridging a wide gap between theory and practice.

This may also be considered as a reference which presents the state-of-the-art research on portfolio construction with multiple and complex investment objectives and constraints. The book consists of eight chapters. In Chapter 1, we provide a brief introduction. In Chapter 2, we present the fundamental issues of modern portfolio theory. In Chapter 3, the various multicriteria decision aid methods, either discrete or continuous, are concisely described. In Chapter 4, we present a comprehensive literature review on the field of multicriteria portfolio management. In Chapter 5, we develop an integrated and original multicriteria portfolio construction methodology. In Chapter 6, we present the web-based information system, in which the suggested methodological framework has been implemented. In Chapter 7, we discuss the experimental application of the proposed methodology and in Chapter 8, we provide the overall conclusions.

We consider the highlight of the book to be the detailed and step-by-step implementation of the proposed multicriteria algorithms in Python programming language. The implementation has been carried out in detail; each step is elaborately described, from the input of the data to the extraction of the results. Algorithms are organized into small cells of code, accompanied by targeted remarks and comments, in order to help the reader understand their mechanics. In addition, readers are provided with access to the source code through GitHub.

The target audience of the book includes a very diversified group of readers, such as fund managers, risk managers, investment advisors, bankers, sophisticated private investors, as well as analytics scientists, operations research scientists, and computer engineers. Finally, the book can be used as a textbook for either advanced undergraduate or postgraduate courses in investment analysis, portfolio engineering, and computer science for financial engineering.

Athens, Greece Elissaios Sarmas

Paris, France Panos Xidonas

Athens, Greece Haris Doukas
June 9, 2020

Contents

Chapter 1
Introduction

Nowadays, one of the major problems of the financial sector is the creation and management of an efficient investment portfolio under the complex environment of globalized society, rapidly increasing competition and sweeping economic changes at national and international level. In general, an investment portfolio is a basket of assets, the selection of which was based on specific economic objectives and constraints, aiming at the generation of profits for the investor.

Until the 1950s, the concept of portfolios was completely different. Investing in equities was a stochastic process as there was insufficient financial data available and few people had realized the importance of investment management. Investors usually focused on the opportunities offered by each equity and not in a return-risk relationship.

The above situation radically changed since 1952, when Nobel laureate H. Markowitz published his research work under the title "Portfolio Selection" [112], where he introduced the mathematical relation between the return and the risk of a security. According to Markowitz mean–variance model, a combination of different kinds of equities is less risky than owning only one type. Within this new reality, investors started creating portfolios that favored specific investment styles and preferences, using the mean–variance model or other models, which tried to expand it and extinguish its weaknesses. Therefore, nowadays the process of creating and managing equity portfolios has significantly developed and cultivated.

However, it is obvious that the global economy has historically been shaken by strong fluctuations, making equities one of the most vulnerable markets. Equity portfolios are the riskiest market placement for two main reasons: Firstly, there is no possibility of differentiating part of the risk, investing in fixed-income securities and deposit or derivative products. Secondly, the process of equity portfolio management is extremely difficult due to the existence of a large number of equities traded on stock markets. This fact renders necessary the investigation of thousands of securities, which are available as investment choices.

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_1

Equity portfolio management is a very complex problem, as it focuses on three different levels of decision-making: (i) selecting equity securities which encapsulate the best investment prospects, (ii) distributing the available capital in order to achieve optimal portfolio composition, and (iii) comparative evaluation of the constructed portfolios. Besides, the problem of equity portfolio management is linked to three other fundamental parameters that affect each decision-making process: uncertainty, the existence of multiple criteria, and the profile and preferences of the decision-maker (DM)

Last but not least, another crucial parameter is the existence of many stakeholders because of the complexity of modern economies and markets. In fact, the whole process is made more difficult by the fact that these stakeholders usually have different, or even conflicting, interests. More specifically, the entities that constitute the environment of this problem, can be grouped into four categories: (a) entities which are associated with the supervision of the market, (b) companies listed on the stock market, (c) institutional and private investors and (d) investment service providers.

In conclusion, the above parameters demonstrate the enormous complexity and uncertainty in the financial decision-making process and imply the need for appropriate indicators and supportive decision tools. These tools are intended to replace decision-making based exclusively on empirical approaches with modern methods of analysis, resulting in a more efficient treatment of investment risks and equity portfolios management.

This book provides an integrated multicriteria methodology for portfolio management, which aspires to address the problem of portfolio construction and optimization, taking into consideration the limits related to the Markowitz mean–variance model and the behavior of the decision-maker, who could potentially have additional criteria in mind, beyond risk and return. Thus, the proposed methodology treats the portfolio management problem as a 2-phase process: (a) The phase of multicriteria portfolio selection and (b) the phase of multiobjective portfolio optimization.

In this book, we present the pseudocode for a series of multicriteria decision analysis methods, which are used to address the problem of portfolio construction and selection, along with several small-scale numerical examples. In this context, the reader can become familiar with the algorithms used very quickly. Additionally, we address the problem of portfolio optimization with the development of four individual multiobjective methodologies: (a) the classic mean–variance methodology, equipped with a complete series of policy constraints, (b) the goal programming methodology, (c) the multiobjective programming methodology, which includes the optimization of PROMETHEE net flow, and (d) the genetic algorithm methodology for portfolio optimization.

Finally, a major contribution of the book is the detailed, step-by-step implementation of the proposed algorithms in Python programming language. More specifically, a series of multicriteria decision analysis methods and multiobjective optimization methods are efficiently implemented in Python, providing extensive solutions for a wide range of MCDA problems. The implementation has been carried out in detail, as each step is carefully described, from the input of the data to the extraction of the results. The algorithms are organized into small cells of code accompanied by targeted observations and comments, in order to help the reader understand their mechanics. In addition, readers are provided with access to the source code, which includes the numerical examples discussed in the book, in the following link: https://github.com/epu-ntua/Multicriteria-Portfolio-Construction-with-Python:

The book proceeds as follows. In Chapter 2 we present the fundamental issues of the portfolio management process. In Chapter 3, the multicriteria decision analysis methods are concisely described. In Chapter 4 we present a literature review of the developed MCDA methodologies for portfolio management. In Chapter 5 we introduce the proposed methodology. Chapter 6 presents the web-based information system, in which the suggested methodological framework is implemented. Additionally, the utilized algorithms implementation is explained with extensive use of the Python language. In Chapter 7 we present the experimental application of the proposed methodology. Finally, the conclusions are provided in Chapter 8.

Chapter 2
The Portfolio Management Problem

2.1 Introduction

This chapter focuses on the portfolio management problem, and more specifically, the mean–variance model introduced by Harry Markowitz [112, 113]. The problem of portfolio optimization was introduced as a quadratic mathematical programming problem. Since then, the development in this field was rapid, as many scientists have attempted to improve this methodology and cure its weaknesses [60]. In discussing the various portfolio theory techniques, we will adhere to standard notation such as in Xidonas et al. [222]. The presentation of the mean–variance methodological framework is developed in three sections.

In Section 2.2 there is a brief introduction to the basic formulas that constitute the problem. The most significant terms are defined, such as the return and the risk both in case of a single security, as well as in the general case of a portfolio of securities.

In Section 2.3 the fundamental principle of diversification is presented. We introduce how diversification affects the portfolio risk, analyzing the factors that make necessary the endorsement of a diversified strategy. In this section the analysis is divided into two parts: in the first part we consider the case of uncorrelated securities, while in the second part we consider the most realistic case of correlated securities.

In Section 2.4 there is a detailed description of the problem of portfolio optimization. The concepts of efficient portfolios and efficient frontier are introduced. Both the case that short sales are allowed and the case that short sales are restricted are discussed. Finally, the concept of the risk-free security is introduced.

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_2

2.2 Return and Risk

A legal contract which allows the investor to receive some future economic benefits under specific and clearly formulated conditions is called *security*. *Common stocks* or *equities* are a subcategory of securities which provide the investor the right to participate in the profits of the company.

Let r_{ij} be the return of a security i during a time period j, then the expected return $E(r_i)$ of the security i, concerning a series of M future time periods which can be considered equally important, is defined as follows:

$$E(r_i) = \sum_{j=1}^{M} \frac{r_{ij}}{M} \tag{2.1}$$

Additionally, if the future returns of a security cannot be considered equally important, then the expected return can be calculated with the following formula:

$$E(r_i) = \sum_{j=1}^{M} P_{ij} r_{ij} \tag{2.2}$$

where P_{ij} is the possibility of the return j for the security i.

The most common metric of a security's risk is the variance of returns. Let r_{ij} be the return of a security i during a time period j and $E(r_i)$ be the expected return of the security, then the variance of returns σ_i^2 of the security i, concerning a series of M future time periods which can be considered equally important, is defined as follows:

$$\sigma_i^2 = \sum_{j=1}^{M} \frac{(r_{ij} - E(r_i))^2}{M} \tag{2.3}$$

If the future returns of a security cannot be considered equally important, then the variance of returns can be calculated as follows:

$$\sigma_i^2 = \sum_{j=1}^{M} P_{ij}(r_{ij} - E(r_i))^2 \tag{2.4}$$

where P_{ij} is the possibility of the return j for the security i.

The term *portfolio* refers to any combination of financial assets such as stocks, bonds, and cash. Each asset participates in the portfolio in some proportion which is determined by the value of the asset relatively to the total value of the portfolio. In the following section, the concept of security portfolios is discussed.

Let us consider a portfolio P which include N securities. The j-return of this portfolio is defined as follows:

$$r_{Pj} = \sum_{i=1}^{N} w_i r_{ij} \tag{2.5}$$

where w_i is the portfolio percentage and r_{ij} is the j-return of security i. The portfolio expected return can be calculated as follows:

$$E(r_P) = E\left(\sum_{i=1}^{N} w_i r_{ij}\right) = \sum_{i=1}^{N} E(w_i r_{ij}) = \sum_{i=1}^{N} w_i E(r_i) \tag{2.6}$$

where w_i is the portfolio percentage and $E(r_i)$ is the expected return of security i.

Finally, the portfolio risk σ_i^2 can be defined as follows:

$$\sigma_P^2 = \sum_{i=1}^{N} w_i^2 \sigma_i^2 + \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} w_i w_j \sigma_{ij} \tag{2.7}$$

where σ_{ij} is the covariance between securities i and j and w_i is the participation percentage of security i in the portfolio.

In vector form, portfolio return and risk can be expressed with the following equations:

$$E(r_P) = \mathbf{r}^T \mathbf{w} \tag{2.8}$$

$$\sigma_P^2 = \mathbf{w}^T \mathbf{V} \mathbf{w} \tag{2.9}$$

where **V** is the variance–covariance matrix, **r** is the return column vector, and **w** is the weighting factor vector.

The covariance between the returns of securities i and j is defined as follows:

$$\sigma_{ij} \doteq \frac{1}{M} \sum_{t=1}^{M} [r_{it} - E(r_i)][r_{jt} - E(r_j)] \tag{2.10}$$

2.3 Portfolio Diversification

As shown in the previous section, the main components which were originally used for portfolio analysis are *portfolio expected return* and *portfolio risk*. The difference between these components lies in the fact that portfolio expected return is a linear function of the individual securities' return. On the other side, portfolio risk is a nonlinear function of the securities' risk, as it is represented with the covariance matrix. In this section we try to analyze the risk function in the simplistic case of uncorrelated securities, as well as in the general case of correlated securities.

2.3.1 Uncorrelated Securities

Firstly, let us consider the case of a portfolio consisting of uncorrelated securities, meaning that $\sigma_{ij} = 0$ for each pair of securities i and j. In this case the portfolio risk of a portfolio including N securities is

$$\sigma_P^2 = \sum_{i=1}^{N} (w_i^2 \sigma_i^2) \tag{2.11}$$

Assuming that the available capital can be equally distributed among the securities ($w_1 = w_2 = \cdots = w_N = 1/N$), then the portfolio risk is calculated as follows:

$$\sigma_P^2 = \sum_{i=1}^{N} \left(\frac{1}{N}\right)^2 \sigma_i^2 = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\sigma_i^2}{N}\right) = \frac{1}{N} \bar{\sigma}_i^2 \tag{2.12}$$

where $\bar{\sigma}_i^2$ represents the mean variance of the portfolio's securities returns.

Thus, we conclude that as the number of securities N which participate to the portfolio increases, the portfolio risk is reduced. Therefore, in case of a well-diversified portfolio, the portfolio risk can be significantly limited. However, the hypothesis of uncorrelated securities among the securities is not realistic.

2.3.2 Correlated Securities

Now, let us consider the general case where $\sigma_{ij} \neq 0$. Assuming that the available capital can be equally distributed among the securities, then the portfolio risk can be calculated as follows:

$$\begin{aligned} \sigma_P^2 &= \sum_{i=1}^{N} \left(\frac{1}{N}\right)^2 \sigma_i^2 + \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \left(\frac{1}{N}\right)^2 \sigma_{ij} \\ &= \frac{1}{N} \left(\sum_{i=1}^{N} \frac{\sigma_i^2}{N}\right) + \frac{N-1}{N} \left(\sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \frac{\sigma_{ij}}{N(N-1)}\right) \end{aligned} \tag{2.13}$$

As stated in the previous paragraph, the first term inside the parenthesis represents the mean variance of the portfolio's securities returns. The second term inside the parenthesis represents the mean covariance $s i g \bar{m} a_{ij}$ of the portfolio's securities returns. Consequently, the portfolio risk could be expressed as follows:

$$
\begin{aligned}
\sigma_P^2 &= \frac{1}{N}\bar{\sigma}_i^2 + \frac{N-1}{N}\bar{\sigma}_{ij} \\
&= \frac{1}{N}(\bar{\sigma}_i^2 - \bar{\sigma}_{ij}) + \bar{\sigma}_{ij}
\end{aligned}
\tag{2.14}
$$

The previous equations show the effect of diversification at the portfolio risk. If the number of securities becomes very large, then each individual security's risk can be eradicated. In this case, the portfolio variance is minimized, becoming equal to mean variance of the portfolio's securities returns.

2.4 Calculating Efficient Frontiers

The observations made in the previous section lead to the conclusion that a diversification strategy is necessary for the construction of a solid portfolio. The first major methodological framework was developed by H. Markowitz in 1952 [112] introducing the concepts of the *efficient portfolio* and the *efficient frontier*. According to Markowitz definition, a portfolio P is efficient if and only if there is no other portfolio P' such that $E(r_{P'}) \geq E(r_P)$ and $\sigma_{P'} \leq \sigma_P$, given that at least one inequality is strict. Thus, a portfolio P is efficient if and only if there is no other portfolio P' which outweighs P either concerning return or risk. The set including all the efficient portfolios is called *efficient frontier*.

As indicated in Figure 2.1, a feasible portfolio is any portfolio with proportions summing to one. The set of all feasible portfolios is called feasible set and it is depicted as the area inside and to the right of the curved line. All portfolios which have minimum variance for a given mean return are called envelope portfolios and they are depicted on the envelope of the feasible set [19]. Finally all portfolios which have maximum return given the portfolio variance are called efficient portfolios and are depicted by the curve in Figure 2.1.

In case that short sales are not allowed, the efficient frontier is a curve which extends from the *minimum variance portfolio (mV)* to the *maximum return portfolio (MR)*. In any case, the efficient frontier is a concave function which does not have any convex parts. The previous claim can be proved if we consider that it is impossible for a combination of securities (or portfolios) to result in greater risk compared to the risk expressed by the line which connects these securities (or portfolios).

Short sales concern the situations where the investor can sell a security which he does not own. Short sales are made when the investor considers that the security value will decrease; thus, he can benefit by selling the security today in a higher price than he will repurchase it in the future. This strategy enables the construction of portfolios which do not have a finite return upper bound. Therefore, in case that short sales are allowed, the efficient frontier is a curve which extends from the minimum variance portfolio (mV) to infinity, as no return upper bound exists.

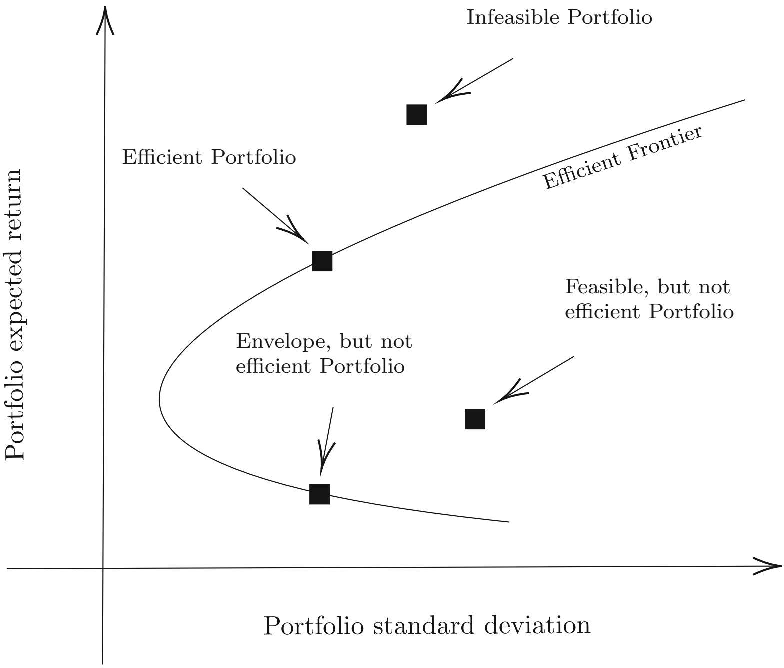


Fig. 2.1 Visualization of the efficient frontier

Finally, let us consider the case that the investor has the opportunity to select the risk-free security. A security which has a certain rate of return, free of the various possible sources of risk, is called *risk-free security*. Given the existence of a risk-free security, the investor can compose a portfolio FP which combines the risk-free security F with a risk portfolio P constituted from a set of other securities.

Let r_F be the return of the risk-free security and $E(r_P)$, σ_P^2 the expected return and the risk of the risk portfolio, respectively. Additionally, let us consider $\sigma_F^2 = 0$ for the risk-free security and zero correlation between the risk portfolio and the risk-free security ($\rho_{FP} = 0$). The portfolio return is given by the following equation:

$$r_{FP} = w_P E(r_P) + (1 - w_P) r_F \tag{2.15}$$

The portfolio standard deviation is defined as follows:

$$\begin{aligned} \sigma_{FP} &= \sqrt{w_P^2\sigma_P^2 + (1-w_P)^2\sigma_F^2 + 2w_P(1-w_P)\rho_{FP}\,\sigma_P\,\sigma_F = w_P^2\sigma_P^2} \\ &= \sqrt{w_P^2\sigma_P^2} \\ &= w_P\sigma_P \Rightarrow \\ w_P &= \frac{\sigma_{FP}}{\sigma_P} \end{aligned} \tag{2.16}$$

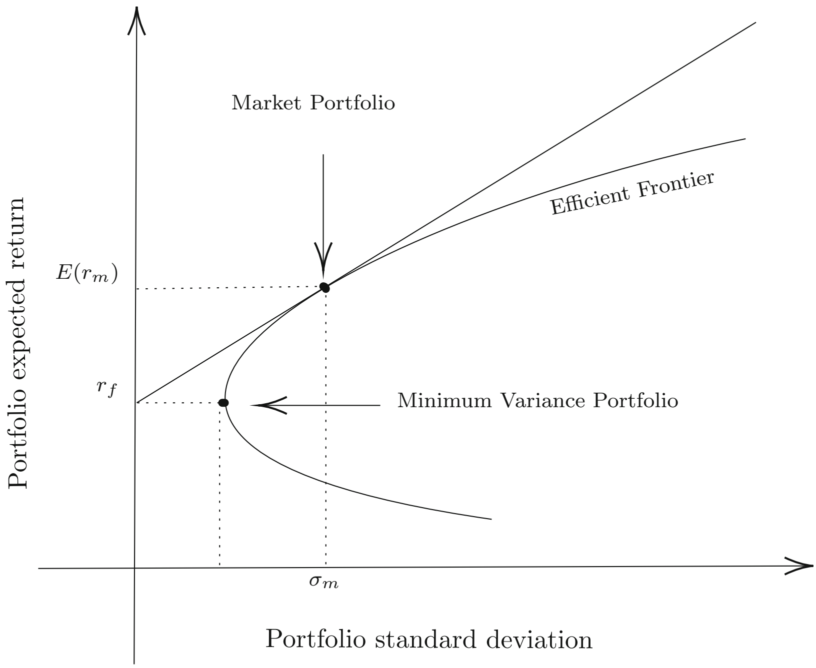


Fig. 2.2 Visualization of the efficient frontier with risk-free security

Thus, the portfolio return is

$$
\begin{aligned}
r_{FP} &= (1 - w_P) r_F + w_P E(r_P) \qquad \Rightarrow \\
r_{FP} &= (1 - \frac{\sigma_{FP}}{\sigma_P}) r_F + \frac{\sigma_{FP}}{\sigma_P} E(r_P) \quad \Rightarrow \\
r_{FP} &= r_F + \frac{E(r_P) - r_F}{\sigma_P} \sigma_{RF}
\end{aligned}
\tag{2.17}
$$

The portfolio return is a linear function of risk, resulting to the formulation of a new efficient frontier. As shown in Figure 2.2, in case that there is a risk-free security, the efficient portfolios are placed in the line that intersects the vertical axis at point r_F. Consequently, there is only one optimal portfolio FP, which is determined by the risk portfolio P placed in the tangent of the above equation with the risk portfolios efficient frontier.

The portfolio P includes m risk securities with weighting factors $w_1, w_2, \ldots, w_m$ and the risk-free security with weighting factor w_F, such that $w_1 + w_2 + \cdots + w_m + w_F = 1$. The problem lies to the definition of the proportions of the portfolio FP.

Having described the relation between the portfolio return and risk, we present the appropriate techniques for the determination of the efficient frontier for the following cases [217, 222]:

1. Short sales are allowed and there is possibility of investment to a risk-free security
2. Short sales are allowed and there is no possibility of investment to a risk-free security
3. Short sales are not allowed and there is no possibility of investment to a risk-free security
4. Short sales are not allowed and there is possibility of investment to a risk-free security

2.4.1 Short Sales Allowed and Risk-Free Security

The risk-free security option results to the existence of a portfolio which dominates all other feasible portfolios. The slope of the line that relates the risk-free security to the risk portfolio is equal to the excess return of the portfolio divided to its volatility (Figure 2.3). The excess return of a portfolio represents the difference between the portfolio expected return and the risk-free security return. Therefore, the aim is to maximize the slope and the problem is formulated as follows:

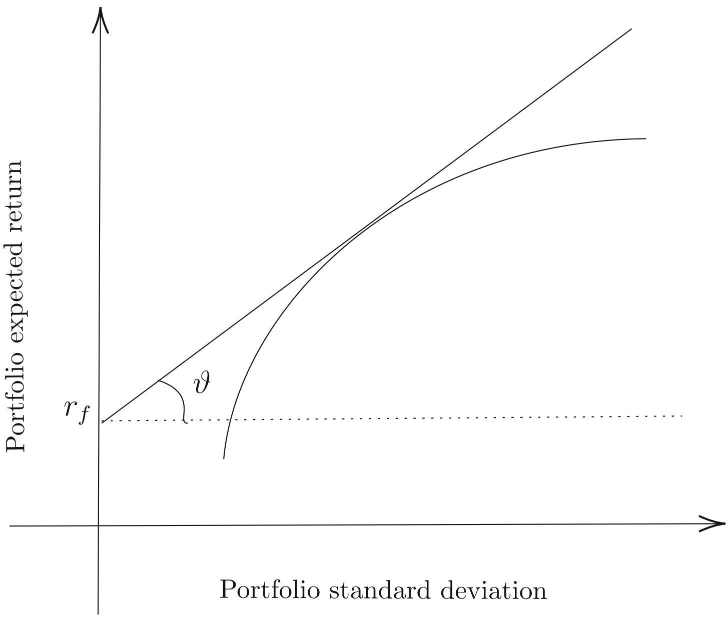


Fig. 2.3 Maximization of the envelope θ for the determination of the efficient frontier given a risk-free security

$$
\begin{aligned}
&\max_{w} \theta = \frac{E(r_P) - r_F}{\sigma_P} \\
&\text{s.t.} \sum_{i=1}^{m} w_i = 1 \\
&\qquad w_i \in \mathbb{R} \qquad i = 1, 2, \ldots, m
\end{aligned}
\tag{2.18}
$$

The only restriction of the problem is incorporated in the objective function. Thus, the problem is transformed to a maximization problem with no additional restrictions. The proof [222] is given below:

Proof Given that:

$$
r_F = \left(\sum_{i=1}^{m} w_i \right) r_F = \sum_{i=1}^{m} (w_i r_F)
$$

the objective function takes the following form:

$$
\theta = \frac{\sum_{i=1}^{m} w_i (r_i - r_F)}{\left(\sum_{i=1}^{m} w_i^2 \sigma_i^2 + \sum_{i=1}^{m} \sum_{j=1,\ j \neq i}^{m} w_i w_j \sigma_{ij} \right)^{1/2}}
\tag{2.19}
$$

The weighting factors w_k of the invested capital—which maximize the objective function θ can be determined solving the following system of partial derivatives $\partial \theta / \partial w_k$ if they are set equal to zero:

$$
\begin{aligned}
&\frac{\partial \theta}{\partial w_k} = 0 \Rightarrow \\
&(r_k - r_F) \left(\sum_{i=1}^{m} w_i^2 \sigma_i^2 + \sum_{i=1}^{m} \sum_{j=1,\ j \neq i}^{m} w_i w_j \sigma_{ij} \right)^{-1/2} + \left[\sum_{i=1}^{m} w_i (r_i - r_F) \right] \\
&\left[-\frac{1}{2} \left(\sum_{i=1}^{m} w_i^2 \sigma_i^2 + \sum_{i=1}^{m} \sum_{j=1,\ j \neq i}^{m} w_i w_j \sigma_{ij} \right)^{-3/2} \left(2 w_k \sigma_k^2 + 2 \sum_{j=1,\ j \neq k}^{m} w_j \sigma_{kj} \right) \right] = 0 \Rightarrow \\
&(r_k - r_F) - \left(\frac{\sum_{i=1}^{m} w_i (r_i - r_F)}{\sum_{i=1}^{m} w_i^2 \sigma_i^2 + \sum_{i=1}^{m} \sum_{j=1,\ j \neq i}^{m} w_i w_j \sigma_{ij}} \right) \left(w_k \sigma_k^2 + \sum_{j=1,\ j \neq k}^{m} w_j \sigma_{kj} \right) = 0
\end{aligned}
\tag{2.20}
$$

Let λ be

$$\lambda = \frac{\sum_{i=1}^{m} w_i (r_i - r_F)}{\sum_{i=1}^{m} w_i^2 \sigma_i^2 + \sum_{i=1}^{m} \sum_{j=1,\ j \neq i}^{m} w_i w_j \sigma_{ij}} \tag{2.21}$$

the last equation gives

$$\begin{aligned} (r_k - r_F) - \lambda \left(w_k \sigma_k^2 + \sum_{j=1,\ j \neq k}^{m} w_j \sigma_{kj} \right) &= 0 \\ r_k - r_F = \lambda w_k \sigma_k^2 + \sum_{j=1,\ j \neq k}^{m} \lambda w_j \sigma_{kj} \end{aligned} \tag{2.22}$$

Let the new variable $Z_k = \lambda w_k$, the last equation gives for every variable i:

$$r_i - r_F = Z_1 \sigma_{1i} + Z_2 \sigma_{2i} + \cdots + Z_i \sigma_i^2 + \cdots + Z_m \sigma_{mi} \tag{2.23}$$

thus, formulating the following linear system:

$$\left\{ \begin{array}{l} r_1 - r_F = Z_1 \sigma_1^2 + Z_2 \sigma_{21} + \cdots + Z_m \sigma_{m1} \\ r_2 - r_F = Z_1 \sigma_{12} + Z_2 \sigma_2^2 + \cdots + Z_m \sigma_{m2} \\ \quad \cdots\cdots \\ r_m - r_F = Z_1 \sigma_{1m} + Z_2 \sigma_{2m} + \cdots + Z_m \sigma_m^2 \end{array} \right\}$$

The weighting factors w_k of the invested capital are obtained solving the above linear system using the following equation:

$$w_k = \frac{Z_k}{\sum_{i=1}^{m} Z_i} \tag{2.24}$$

□

2.4.2 *Short Sales Allowed and No Risk-Free Security*

In this case, short sales are allowed, thus not constraining the weighting factors to be positive. The problem is defined as follows:

$$
\begin{aligned}
&\min_{\mathbf{w}} \sigma_P^2 = \mathbf{w}^T \mathbf{V} \mathbf{w} \\
&\max_{\mathbf{w}} E(r_P) = \mathbf{r}^T \mathbf{w} \\
&\text{s.t. } \mathbf{e}^T \mathbf{w} = 1 \\
&\qquad \mathbf{w} \in \mathbb{R}
\end{aligned}
\tag{2.25}
$$

In case that short sales are allowed and there is no possibility to invest to a risk-free security, the efficient frontier can be determined applying the methodology that we presented in the previous paragraph. For various values of the risk-free security return, which hypothetically exists, the corresponding efficient portfolios are calculated, until the whole efficient frontier is traced.

2.4.3 No Short Sales Allowed and Risk-Free Security

In case that short sales are not allowed, the problem is similar to the one with short sales allowed with the risk-free security existence, but an additional restriction concerning the weighting factors is added ($w_i \geq 0$). The definition of the problem is presented below:

$$
\begin{aligned}
&\max_{w} \theta = \frac{E(r_P) - r_F}{\sigma_P} \\
&\text{s.t. } \sum_{i=1}^{m} w_i = 1 \\
&\qquad w_i \geq 0 \qquad i = 1, 2, \ldots, m
\end{aligned}
\tag{2.26}
$$

This is a quadratic programming problem with linear restrictions. The solution of this problem can be given using algorithms based on the *Kuhn–Tucker* conditions. These conditions secure that if a solution is found, then this solution is guaranteed to be optimal. The Kuhn–Tucker conditions are presented as follows:

$$
\begin{aligned}
\frac{\partial \theta}{\partial w_i} + U_i &= 0 \\
w_i U_i &= 0 \\
w_i &\geq 0 \\
U_i &\geq 0
\end{aligned}
\tag{2.27}
$$

Any solution that satisfies the previous conditions results in a portfolio which belongs to the efficient frontier.

2.4.4 No Short Sales Allowed and No Risk-Free Security

The general approach assumes that there is a short sales restriction, thus allowing the portfolio proportions vary in range [0, 1]. The original portfolio optimization problem is formulated as follows:

$$
\begin{aligned}
&\max_{w}\ E(r_P) = \sum_{i=1}^{m} w_i E(r_i) \\
&\min_{w}\ \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} w_i w_j \sigma_{ij} \\
&\text{s.t.} \sum_{i=1}^{m} w_i = 1 \\
&\qquad w_i \geq 0, \qquad\qquad i = 1, 2, \ldots, m
\end{aligned}
\tag{2.28}
$$

In matrix form the problem is formulated as follows:

$$
\begin{aligned}
&\max_{\mathbf{w}}\ E(r_P) = \mathbf{r}^T \mathbf{w} \\
&\min_{\mathbf{w}}\ \sigma_P^2 = \mathbf{w}^T \mathbf{V} \mathbf{w} \\
&\text{s.t.}\ \mathbf{e}^T \mathbf{w} = 1 \\
&\qquad \mathbf{w} \geq 0
\end{aligned}
\tag{2.29}
$$

where $\mathbf{V}$ is the variance–covariance matrix, $\mathbf{r}$ is the return column vector, $\mathbf{e}$ is the unary vector, and $\mathbf{w}$ is the weighting factor vector.

This is a quadratic programming problem with linear restrictions. The specification of the efficient frontier can be achieved by solving the problem of risk minimization, parametrically on the expected portfolio return, as follows:

$$
\begin{aligned}
&\min_{\mathbf{w}}\ \sigma_P^2 = \mathbf{w}^T \mathbf{V} \mathbf{w} \\
&\text{s.t.}\ \mathbf{r}^T \mathbf{w} = R \\
&\qquad \mathbf{e}^T \mathbf{w} = 1 \\
&\qquad \mathbf{w} \geq 0
\end{aligned}
\tag{2.30}
$$

where parameter R is the predefined expected portfolio return. Varying the expected return R between the edge values of *minimum variance portfolio* and *maximum return portfolio*, results in the specification of the whole efficient frontier.

2.5 Conclusions

The complexity of most decision problems requires the introduction of multiple and often conflicting criteria. However, Markowitz model is based on two criteria, failing to incorporate additional goals to performance and risk, that an investor can set in a realistic framework and thus limiting the participation of the decision-maker in the investment process [185]. Additionally, the possibilities for accurately determining the investor's risk tolerance profile and for expressing his special preferences are rather limited [80]. An integrated approach to the portfolio management problem requires thorough consideration of all the criteria that affect the course of stock markets, such as financial data (profitability indicators, liquidity, etc.) and stock market indicators (dividend yield, systematic risk, etc.). This observation leads to the conclusion that the classical methodological approach is rather inadequate, emphasizing the need for more flexible methods.

Chapter 3
Multicriteria Decision Analysis Methods

3.1 Introduction

The discussion of the portfolio management problem in the previous chapter signified the need for new methodological decision support frameworks, in order to overcome the existing problems and cure the inadequacies of the conventional mean–variance model.

In this chapter, there is a short introduction to multicriteria decision analysis, as it is the most appropriate field to support the portfolio management decision-making process, according to Xidonas et al. [217, 223].

In the first section there is a presentation of the basic concepts of this scientific field, as well as a general methodology overview, analyzing the four phases of decision support. Additionally, an introduction to discrete multicriteria decision support methods is made, presenting and comparing the three basic sectors (multiattribute utility theory, outranking relations theory, preference disaggregation approach).

In the second section the continuous optimization methods are developed. This section begins with the fundamental concepts of linear, quadratic, and integer programming. In the following paragraph there is an introduction to multiobjective programming problems. Afterwards, the methodological framework of goal programming problems is presented, followed by an introduction to genetic algorithms.

3.2 Discrete Multicriteria Decision Analysis

Multicriteria decision analysis (MCDA) or Multicriteria decision-making (MCDM) belongs to the scientific field of operational research. The main objective of all methodological approaches in the field of multicriteria decision analysis is the

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_3

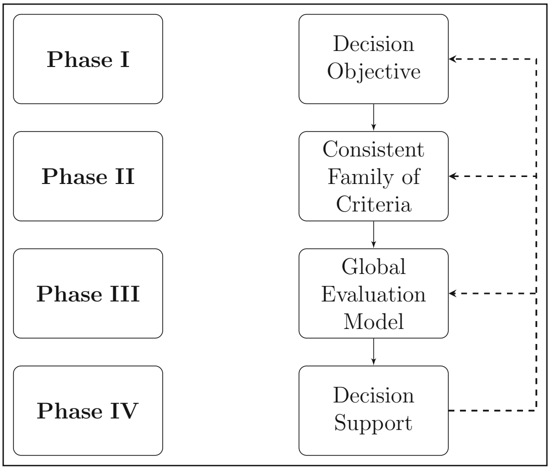


Fig. 3.1 Multicriteria decision analysis methodological framework (Roy, [157])

development of models that incorporate all the parameters of the problem in order to support the decision-maker in the decision-making process.

3.2.1 Basic concepts and methodology

In 1985, Roy, one of the founders of multicriteria analysis modern theory, presented a general methodological framework for multidimensional decision-making problems [157]. As shown in Figure 3.1, the analysis process of multicriteria decision-making problems involves four stages among which feedback can be developed.

Phase 1: Decision Objective

In this phase, there are two basic tasks which are necessary to be completed: (a) Strict definition of set A of alternatives or actions of the problem and (b) identification of the decision problematic.

The set A of the alternatives of the problem could be a continuous set or a discrete set. In the case of a continuous problem, the continuous set of solutions is defined by mathematical equations (linear inequalities) as a super-hydride with as many dimensions as the multitude of the decision variables. In the case of a discrete problem, the set of feasible solutions is defined by the exhaustive enumeration of its elements.

The decision problematic determines the way that alternatives should be examined. According to Roy [157], there are four main categories of discrete problems (Figure 3.2):

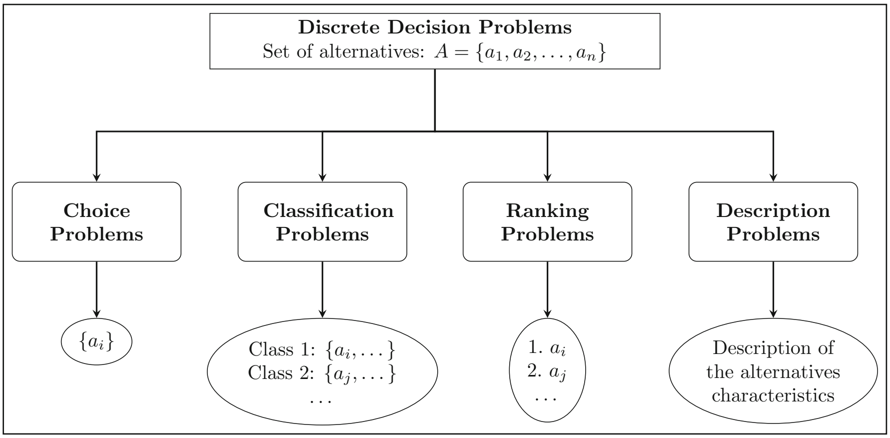


Fig. 3.2 Discrete decision problematics

i. Choice problems refer to the situation where the DM must choose the most suitable alternatives.
ii. Classification problems refer to the situation where the alternatives must be classified into predefined classes.
iii. Ranking problems refer to the situation where the alternatives should be ranked in decreasing order.
iv. Description problems refer to the situation where the alternatives are described according to their performance in individual criteria.

Phase 2: Consistent Family of Criteria

Each factor that affects a decision is a criterion. Formally, a criterion is a monotonic function f which declares the preference of the decision-maker, so as for any two alternatives x_i, x_j the following equations apply

$$f(x_i) > f(x_j) \Leftrightarrow x_i \succ x_j \tag{3.1}$$

$$f(x_i) = f(x_j) \Leftrightarrow x_i \sim x_j \tag{3.2}$$

where the notation $x_i \succ x_j$ declares that alternative x_i is preferable to x_j and the notation $x_i \sim x_j$ declares that there is indifference between the two alternatives.

This procedure results in the configuration of a consistent family of criteria. A set of criteria $F = \{f_1, \ldots, f_q\}$ configures a consistent family of criteria, if and only if the following properties are satisfied:

1. **Monotonicity:** A set of criteria is monotonic if and only if for any two alternatives x_i, x_j such that $f_k(x_i) > f_k(x_j)$ for any criterion k and $f_l(x_i) = f_l(x_j)$ for any other criterion $l \neq k$, it is concluded that $x_i \succ x_j$.
2. **Exhaustivity:** A set of criteria is exhaustive if and only if for any two alternatives x_i, x_j such that $f_k(x_i) = f_k(x_j)$ for any criterion k, it is concluded that $x_i \sim x_j$.
3. **Non-redundancy:** A set of criteria is non-redundant if and only if the removal of any criterion leads to monotonicity or exhaustivity property violation.

Phase 3: Global Evaluation Model

The global evaluation model is defined as the composition of all the criteria, in order to analyze the problem according to the determined problematic. The global evaluation model can be applied to determine a total evaluation of the alternatives, to explore the solution set (for continuous problems) and to execute pairwise comparisons between all pairs of alternatives.

Phase 4: Decision Support

This phase of the process involves all the activities which help the decision-maker understand the results of the application of the model. The role of the consultant is of crucial importance because he must organize the answers in a comprehensible way.

There are three main fields which deal with discrete multicriteria decision problems: (a) *Multiattribute utility theory*, (b) *Outranking relation theory*, and (c) *Preference disaggregation approach.*

3.2.2 Multiattribute Utility Theory

Multiattribute utility theory (MAUT) constitutes a generalization of classical utility theory. From the early stages of multicriteria decision analysis, multiattribute utility theory has been one of its fundamental subfields, supporting both its theoretical and its practical evolution.

Multiattribute utility theory uses a *value function* or (utility function) $U(g)$, which represents the value system that the decision-maker follows. This function has the following expression:

$$U(\mathbf{g}) = U(g_1, g_2, \ldots, g_n) \tag{3.3}$$

where $\mathbf{g}$ is the evaluation criteria vector: $\mathbf{g} = g_1, g_2, \ldots, g_n$.

In general, utility functions are non-linear monotonically increasing functions that meet the following properties:

$$U(\mathbf{g}_x) > U(\mathbf{g}_{x'}) \Leftrightarrow x \succ x'$$

and

$$U(\mathbf{g}_x) = U(\mathbf{g}_{x'}) \Leftrightarrow x \sim x'$$

The most widely known form of the utility function is the additive:

$$U(\mathbf{g}) = p_1 u_1(g_1) + p_2 u_2(g_2) + \cdots + p_n u_n(g_n) \tag{3.4}$$

where u_i are the partial utility functions of the evaluation criteria and p_i are the criteria weighting factors, which should sum to one. Each weighting level implies the trade-off that the decision-maker is willing to pay, in order to succeed unary increasement over the corresponding criterion.

The additive utility function is based on the important hypothesis of *mutual preferential independence* of the evaluation criteria, which is explained as follows: A subset g' of the set of evaluation criteria $g' \subset g$ is *preferential independent* of the rest of the criteria, if and only if the preference of the decision-maker about the alternatives, which differ only in terms of the criteria of g', are not affected by the rest of the criteria. The set of the evaluation criteria is considered to fulfill the assumption of mutual preferential independence, if and only if each subset is preferential independent of the rest of the criteria.

The utility function construction process should be based on the cooperation of the decision-maker himself with an expert analyst. The significance level of the evaluation criteria, as well as the form of the partial utility function must be determined before the construction of the utility function. The determination of the partial utility functions is based on interactive techniques, such as direct questions to the decision-maker, which lead to a detailed understanding of the way the decision-maker evaluates the alternatives in each criterion. The best-known technique is called *midpoint value technique*, developed by Keeney and Raiffa [89]. Additionally, various decision support systems have been developed, which implement methods that allow an interactive development and use of the utility functions, such as the MACBETH system introduced by Bana e Costa and Vansnick [48].

A detailed presentation of the multiattribute utility theory, as well as its applications is included in the book of Keeney and Raiffa [89].

The main methods of the multiattribute utility theory are presented in Table 3.1:

3.2.3 Outranking relations theory

The *outranking relations theory* is a special methodological multicriteria analysis sector, which emerged at late 1960s with Bernard Roy's study and the presentation of the *ELECTRE* family methods (ELimination Et Choix Traduisant la Realite) by Roy [154–156] and has widely spread, especially in Europe, since. It must be noted that the outranking relations theory has its roots in the social choice theory developed by Arrow and Raynaud [10].

Table 3.1 Multiattribute utility theory methods

Method	Reference
SAW	MacCrimmon (1968) [109]
AHP	Saaty (1980) [161]
TOPSIS	Hwang and Yoon (1981) [82]
GRA	Julong (1989) [88]
MAUT	Keeney and Raiffa (1993) [89]
MACBETH	Bana e Costa and Vansnick (1994) [48]
Fuzzy TOPSIS	Chen (2000) [41]
ANP	Saaty (2001) [160]
VIKOR	Opricovic and Tzeng (2004, 2007) [135, 136]

Unlike multiattribute utility theory which aims at the development of a utility function, the goal of the outranking relations theory is the development of a methodological framework that allows pairwise comparison between alternatives. All the techniques that are based on the outranking relations theory are applied in two basic phases. The first phase includes the development of an outranking relation between the examined alternatives, while in the second phase the outranking relation is exploited for the evaluation of the alternatives in the desired form (ranking, classification, choice).

The outranking relation **S** is a bilateral relation defined in the set of alternatives, such that:

$$x\mathbf{S}x' \Leftrightarrow \text{alternative } x \text{ is at least as good as alternative } x' \tag{3.5}$$

The idea of outranking relation is that the comparison of two alternatives x and x' is based on the power of both *positive indications*, which support the fact that alternative i is better than alternative j and *negative indications*, which support the opposite fact. In case that the power of positive indications is significant and the power of negative indications is insignificant, we can assume that there is an outranking relation $x\mathbf{S}x'$ between alternatives x and x'.

In fact, the outranking relations theory differs from the multiattribute utility theory in two major points:

- The outranking relation is not transitive. In utility theory the evaluation of the alternatives with the utility function maintains the transitive property. On the contrary, the development and use of outranking relations allows the representation of cases where, while the alternative x_1 is preferable to x_2, and x_2 is preferable to x_3, finally x_1 is neither preferable nor indifferent to x_3.
- The outranking relation is not complete. The completeness property refers to the complete evaluation and ranking of all the alternatives. The multiattribute utility theory leads to a complete evaluation of the alternatives, developing appropriate utility functions. On the other side, the outranking relations theory does not necessarily demand the decision-maker's preferences to carry the transitive

Table 3.2 Multiattribute utility theory methods

Method	Reference
ELECTRE	Roy (1968, 1991, 1996) [154–156]
QUALIFLEX	Paelinck (1976, 1977) [138, 139]
PROMETHEE	Brans et al. (1986) [30]
ORESTE	Roubens (1982) [153]
REGIME	Hinloopen et al. (1983) [75]
EVAMIX	Voogd (1982, 1983), Nijkamp et al. (1990) [131, 205, 206]
MELCHIOR	Leclercq (1984) [103]
TACTIC	Vansnick (1986) [202]
PRAGMA	Matarazzo (1988) [119]
MAPPAC	Matarazzo (1990) [120]
ARGUS	De Keyser and Peeters (1994) [53]
IDRA	Greco (1997) [68]
PACMAN	Giarlotta (1998, 2001) [66]
SIR	Xu (2001) [224]

property, thus a complete evaluation is often impossible. The non-completeness property is very important since the complete ranking of the alternatives is unrealistic in a variety of problems.

Therefore, the *incomparability property* is a property of outranking relations theory that makes it extremely useful for this category of problems. The information provided by the decision-maker is important for the development of the outranking relation. This information is quite different, depending on the method used. However, in most cases it is about (a) the significance of the evaluation criteria (weighting factors) and (b) the preference, indifference, and veto thresholds. These thresholds contribute to the development of a fuzzy outranking relation, where there is partial preference or even indifference among the alternatives.

The main methods of the outranking relation theory are presented in Table 3.2:

3.2.4 Preference Disaggregation Approach

The preference disaggregation approach (Jacquet-Lagreze and Siskos, 1982, 2001) [84, 86] involves the development of a methodological framework which can be used for the analysis of decisions made by the decision-maker, in order to determine the appropriate criteria synthesis model that meets the value system and the preferences of the decision-maker.

The preference disaggregation approach follows a reverse process compared to the multiattribute utility theory and the outranking relations theory. This approach considers that the decision-maker (consciously or not) follows a value system which results in the decisions he makes. It tries to detect the way that decisions are

Table 3.3 Multiattribute utility theory methods

Method	Reference
UTA	Jacquet-Lagreze and Siskos (1982,1983) [84, 85]
UTASTAR	Siskos and Yannacopoulos (1985) [173]
UTADIS	Devaud et al. (1980) [54]
	Jacquet-Lagreze (1995) [83]
	Doumpos and Zopounidis (2002) [58]
MHDIS	Zopounidis and Doumpos (2000) [228]

made and finally reproduce a similar decision-making model. In order to manage to imitate the decision-maker, this method requires a training sample consisting of: (a) a set of decision made by the decision-maker, (b) the evaluation of a set of hypothetical actions, and (c) the evaluation of a subset of the examined alternatives.

The main methods of the preference disaggregation theory are presented in Table 3.3:

3.3 Multiobjective Mathematical Programming

3.3.1 Basic Concepts

A problem that requires to choose the best solution from a set of feasible solutions is called *optimization problem*. These problems are cured by a scientific field which is called *mathematical optimization* (or *mathematical programming*). There are two main categories of optimization problems: Discrete optimization refers to problems with discrete variables and the solution is one or more elements of the feasible set, which is a countable set of possible solutions. Continuous optimization refers to problems with continuous variables and the solution requires to optimize a continuous function subject to one or more equality or inequality constraints. The concept of discrete optimization was discussed in the previous section. In this section, a brief introduction to continuous optimization problems will be made.

The standard form of a continuous optimization problem is the following:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, n \end{aligned} \tag{3.6}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function which should be minimized over a vector x, $g_i(x) \leq 0$ are m inequality constraints, and $h_j(x) = 0$ are n equality constraints, where $m, n \geq 0$. In the above definition, the objective function should be minimized by convention. In case of maximization problem, the objective function should be negated, thus transforming the problem to minimization problem.

In the following paragraphs, some of the major subfields of continuous optimization—which will be used in the methodological framework—will be discussed.

Linear Programming

Linear programming (LP) (Dantzig, 1998) [49] is a continuous mathematical optimization method, which is used in a category of optimization problems, when all the constraints are strictly expressed with linear equations. More specifically, this technique can be used if the objective function of the problem is linear and the requirements of the problem are linear equality or inequality constraints. These linear constraints form a convex polyhedron which is called feasible region of the problem. The linear function is defined on this polyhedron. A linear programming algorithm finds the point where this function has the smallest value (for minimization problem). Some of the most famous applications of linear programming are in the fields of operational research, engineering, scheduling and transportation.

Let x represent a vector of decision variables, b and c represent column vectors of known constants, and A represent a matrix of known constants. The linear problem canonical form is expressed as follows:

$$\begin{aligned} \min_{x} \ & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t. } & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{3.7}$$

The inequalities $Ax \leq b$ and $x \geq 0$ are linear constraints which determine a convex polytope. There are two categories of linear programming algorithms. The first category consists of exchange algorithms, such as the simplex algorithm and the criss-cross algorithm. The simplex algorithm detects a feasible solution at a random vertex of the polyhedron and then walks along the other vertices until the optimal solution is found. The second category consists of interior point algorithms, such as the ellipsoid algorithm, the projective algorithm of Karmarkar, etc.

Quadratic Programming

Quadratic programming (QP) is a continuous mathematical optimization method, which is used for linear constrained quadratic optimization problems. More specifically, the objective function is a quadratic function and the problem restrictions are formed as linear equations and inequations. QP is a specific type of nonlinear optimization.

Let x represent a vector of n decision variables, c a n-dimensional vector, b a m-dimensional vector, Q a $n \times n$ symmetric real matrix, and A a $m \times n$ real vector. The quadratic programming problem with n variables and m constraints is formulated as follows:

$$\begin{aligned} \min_x \ & \tfrac{1}{2}\mathbf{x}^{\mathrm{T}} Q\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} \ & A\mathbf{x} \le \mathbf{b} \end{aligned} \tag{3.8}$$

The solution of quadratic programming problems is particularly simple when Q is positive definite and there are only equality constraints. The solution is produced using Lagrange multipliers and seeking the extremum of the Lagrangian. Therefore given the quadratic problem

$$\begin{aligned} \min_x \ & \tfrac{1}{2}\mathbf{x}^{\mathrm{T}} Q\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} \ & A\mathbf{x} = \mathbf{b} \end{aligned} \tag{3.9}$$

if λ is a set of Lagrange multipliers, the solution is given by the linear system:

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} \end{bmatrix} \tag{3.10}$$

If Q is a positive definite matrix, the appropriate algorithm is the ellipsoid method which provides the solution in polynomial time. Otherwise, the problem is NP-hard, which means that there is no algorithm with polynomial complexity for the problem.

Integer Programming

Integer programming (IP) is a mathematical optimization method for some problems where all the variables are restricted to be integers. In the special case that the objective function and a part of the constraints are linear expressions but the decision variables are integer variables, the appropriate method is called integer linear programming (ILP). Particularly, in case that even some of the decision variables are continuous, the method is called mixed-integer programming (MIP). Finally, in case that both some of the decision variables are continuous and the objective function or some of the constraints are linear functions, then the problem belongs to a subcategory called mixed-integer linear programming (MILP). MILP problems will be part of the proposed methodological framework.

The canonical form of an ILP problem is expressed as:

$$\begin{aligned} \min \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \le \mathbf{b}, \\ & \mathbf{x} \ge \mathbf{0}, \\ \text{and} \quad & \mathbf{x} \in \mathbb{Z}^n \end{aligned} \tag{3.11}$$

where $\mathbf{c}, \mathbf{b}$ are vectors and A is a matrix, under the constraint that all entries are integers.

There are two subcategories of integer programming depending on the range of the decision variables. The first subcategory involves all the problems which demand a decision to be taken. In these problems the integer variables represent decisions with possible answers "yes" or "no," which are translated as 1 or 0. Therefore, problems in which integer variables are restricted in the range {0, 1} are called *zero-one linear programming*. The second subcategory involves problems in which the integer variables represent discrete quantities, i.e. number of discrete pieces of a product. Such problems frequently appear in production planning, scheduling, etc.

In the last decades, a variety of algorithms have been developed, in order to find the optimal solution of an IP problem. The naivest approach is called LP relaxation, recommending to remove all integer constraints, to solve the relaxed LP problem and then to round the solution. However, this technique may find a non-optimal solution or even a non-feasible one if any constraints are violated during the round process. The optimal solution can be found with *cutting plane methods* and *branch and bound* techniques. Finally, another approach to detect an approximate solution to ILP problems is the field of heuristic methods, which do not guarantee an optimal solution but offer better complexity.

3.3.2 Goal Programming

Goal programming (GP) (Charnes and Cooper, 1955, 1961) [38, 39] is a multiobjective optimization technique which extends the concept of linear programming in order to solve problems with multiple conflicting objective functions. The goal programming methodology is quite simple, as each objective function is assigned a goal or target value to be achieved, according to the DM's requirements. Any deviation from this target value is punished with a penalty value. Finally, the weighted sum of all penalty values must be minimized. This technique is useful for the following purposes: (a) The determination of the degree that each goal is fulfilled given the available resources, (b) the estimation of the required resources in order to achieve a predefined goal, and (c) the computation of the best feasible solution under a varying amount of resources and goal priorities.

Let $f_1(x), f_2(x), \ldots, f_k(x)$ be a set of k objective function and $x \in X$, where X is the feasible set of the decision vectors. Let us introduce the *deviational variables* (or slack variables) d_i^-, d_i^+ which represent the amount by which each goal deviates from the target value. More specifically, the vector d^- represents the amount by which each goal's target value is *underachieved*, and vector d^+ represents the amount by which each goal's target is overachieved. Finally, let g_i represent the target value of the i_{th} objective function. The problem is formulated as follows:

$$
\begin{aligned}
\min_{d^+,d^-} \ & \sum_{i=1}^{k} \frac{w_i^+ d_i^+}{g_i} + \frac{w_i^- d_i^-}{g_i} \\
\text{s.t.} \ & f_1(x) + d_1^- - d_1^+ = g_1 \\
& f_2(x) + d_2^- - d_2^+ = g_2 \\
& \qquad \qquad \dots \\
& f_k(x) + d_k^- - d_k^+ = g_k \\
& \qquad \qquad d_i^-, d_i^+ \geq 0 \quad \forall i \in \{1, 2, \dots, k\}
\end{aligned}
\tag{3.12}
$$

where w_i^+ represents the weighting factor of the overachievement penalty of the i_{th} objective function and w_i^- represents the weighting factor of the underachievement penalty of the i_{th} objective function.

In case that underachievement of a goal is undesirable a greater weighting factor w^- is assigned (i.e., $w^- = 1$), else if underachievement is desirable or neutral the weighting factor is set equal to zero ($w^- = 0$). Accordingly, if overachievement of a goal is undesirable a greater weighting factor w^+ is assigned (i.e., $w^+ = 1$), else if overachievement is desirable or neutral the weighting factor is set equal to zero ($w^+ = 0$).

The emerging problem is a linear programming minimization problem which can be easily solved with the methods mentioned above, in order to minimize deviational variables and, subsequently approach the target value of each goal.

3.3.3 Multiobjective Programming

Multiobjective optimization is a sector of MCDA used for mathematical optimization problems that require more than one objective function to be optimized simultaneously.

More specifically, multiobjective linear programming (Steuer, 1986) [183] is an extension of linear programming in case that there are multiple objective functions $f_i(\mathbf{x}) = \mathbf{c}_i^T \mathbf{x},\ i = 1, 2, \dots, k$. The problem is formulated as follows:

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})\} \\
s.t. \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
& \mathbf{x} \geq \mathbf{0}
\end{aligned}
\tag{3.13}
$$

In multiobjective programming, there is not an optimal solution because of the existence of many objective functions, as it is infeasible to optimize all objective functions simultaneously. Subsequently, the concept of the optimal solution is replaced with the concept of an *efficient (or Pareto optimal) solution*, based on dominance theory. In the following paragraphs, the most important definitions of multiattribute theory are introduced (Doumpos, 2009) [56].

Any solution x which satisfies the restrictions of the problem is called *feasible solution*. The set of all feasible solutions is called *feasible set*.

A feasible solution $\mathbf{x}$ is called *Pareto dominant* to another feasible solution $\mathbf{x}'$ if and only if: (a) $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ $\forall i \in \{1, 2, \ldots, k\}$ and (b) $f_j(\mathbf{x}), f_j(\mathbf{x}')$ for at least one $i \in \{1, 2, \ldots\}$.

A solution $\mathbf{x}*$ is called *Pareto optimal*, if and only if there is no other feasible solution that dominates it. The set consisting of all Pareto optimal solutions is called *Pareto frontier* or *Pareto boundary*.

Finally, a solution x is called *weakly optimal* if and only if there is no other feasible solution such that $f_i(\mathbf{x}') \geq f_i(\mathbf{x})$ $\forall i$.

3.3.4 Genetic Algorithms

Evolutionary computation [11, 52] is a scientific subfield of artificial intelligence which is involved with a variety of algorithms for global optimization. These algorithms are population-based trial and error solvers which incorporate meta-heuristic or stochastic optimization characteristics. In biological terms, a population of feasible solutions is subjected to a natural selection process, resulting in a gradual evolution, optimizing the *fitness function* of the problem [62].

Evolutionary algorithms are based on the collective learning process within a population of individuals, each of which represents a search point in the space of potential solutions to a given problem, according to Back and Schwefel [12, 44]. The population is arbitrarily initialized, and it evolves toward better and better regions of the search space with randomized processes of selection, mutation, and recombination. The environment which is called *fitness value* delivers quality information about the search points. The selection process favors those individuals of higher fitness to reproduce more often than those of lower fitness. The recombination mechanism allows the mixing of parental information while passing it to their descendants, and mutation introduces innovation into the population. Evolutionary algorithms are used today in a wide range of sectors [50].

Genetic algorithms (GA) are the most known evolutionary algorithms [78, 209]. The concept of general adaptive processes, concentrating on the idea of a system receiving sensory inputs from the environment by binary detectors, was initially introduced by J. Holland in 1962 [77]. As a result, structures in the search space were modified by operators selected by an adaptive plan, judging the quality of previous trials with an evaluation metric. In a genetic algorithm, a population of individuals is evolved toward better solutions. Each solution has a set of characteristics, called *chromosomes*, which can be mutated and altered. The genetic algorithms generic methodology is presented in Figure 3.3.

In the following paragraphs we present each phase of the methodological framework in detail. However, it is important to give some basic definition before the presentation of each phase. A subset of all the possible solutions to the given problem is called *population*. In every iteration of the algorithm the population

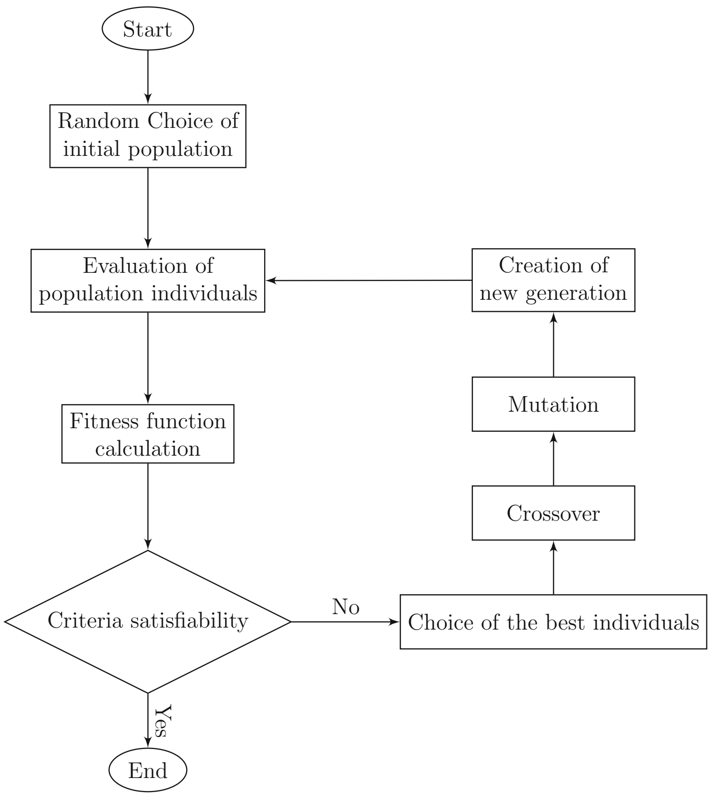


Fig. 3.3 The overall methodological framework of a genetic algorithm

changes, resulting in a different *generation*. Each solution of the population is called *chromosome*. A chromosome is composed of a series of elements called *genes*. There are two basic population models: (a) *Steady State or Incremental* where one or two chromosomes are replaced in each generation and (b) *Generational* where the entire population is replaced in each generation. Additionally, a function which takes a specific solution as input and decides on whether this solution is suitable is called *fitness function*. The algorithm starts with a randomly generated initial population. Then, a series of operators (such as crossover and mutation) are applied on the initial population, in order to generate new individuals. Finally, these new individuals replace the existing individuals in the population and the process repeats. The basic structure of a genetic algorithm is given in Algorithm 1.

In most cases, the individuals of a population are represented with *binary representation*, as it is the simplest representation. Therefore, each individual consists of bit strings. For a variety of problems which demand a binary decision (yes or no), the binary representation naturally helps to structure the individuals. On the other side, for problems that include numbers, these numbers can be easily

Algorithm 1: Genetic Algorithm Standard Form

```
input: popSize (Size of Population), elit (rate of elitism), mut (rate of mutations),
maxIter (maximum number of iterations);
create popSize random feasible solutions
save created solutions in population aray
while ! terminal condition do
    for i in range(maxIter) do
        select the best popsize × elit solutions from population
        save them in popul2
        for j in range(crossover) do
            select two random solutions from population
            generate and save new solutions to popul3
        end
        for j in range(crossover) do
            select a solution from popul3 and mutate with ratemut
            if newSolution.isFeasible() then
                Update popul3 with new solution
            end
        end
    end
    update population = pop2 + pop3
end
Result: the best solution in population
```

transformed to their binary form. Other possible representations which could be used depending on the problem are the *real value representation* and the *integer representation*, though they are not so widely used.

In each generation, the fitness function is evaluated for every individual in the population. The next step involves selecting the best individuals of the population, also known as parent selection. These individuals will recombine to create the individuals for the next generation. The most common technique to choose the best individuals is called *fitness proportionate selection* or *Roulette Wheel Selection* [107]. In fitness proportionate selection, every individual can be chosen with a probability which is proportional to the score of the fitness function. Therefore, individuals with a higher fitting score have a higher possibility to propagate their characteristics to the next generation. Let f_i be the fitness score of individual i in a population of n individuals, then the possibility of the individual i to be selected p_i can be calculated as follows:

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{3.14}$$

Subsequently, the crossover operator is performed, where one or more new individuals are constructed from the previously selected individuals (parents) of the current generation [146]. The crossover operator is usually applied with a probability p_c which is very high. There are many ways to perform the crossover operator in a genetic algorithm. The most used operators are the *one-point crossover*, the *multi-point crossover*, and the *whole arithmetic recombination*. The latter is usually applied for integer representations. The one-point crossover is performed as follows: Let $x = [0, 0, 1, 0, 1]$ and $y = [0, 1, 0, 0, 0]$ be two selected individuals of a population and let us assume that the crossover is defined to happen on the third gene. Therefore, two new individuals will be constructed: $x' = [0, 0, 0, 0, 0]$ and $y' = [0, 1, 1, 0, 1]$. We observe that x' has inherited the first two genes from x and the last three genes from y. Same applies for y', which has inherited the first two genes from y and the last three genes from x.

Finally, mutation is a random change in the chromosome in order to alter the solution [163]. This technique is used in order to increase diversity in the population, thus it is usually performed with a quite low probability p_m. It should be clear that if mutation was applied with a high probability, then it would result in a random search. The simplest mutation operator is the *bit flip mutation*, where some genes are randomly selected and flipped. For example, if we consider a chromosome $x = [1, 1, 0, 0, 1, 0, 1]$ and we apply the mutation operator in its 5_{th} gene, it results in a new mutated chromosome $x' = [1, 1, 0, 0, 0, 0, 1]$.

Thus, each individual is modified and a new generation of individuals is created. The new generation is used as input to the next iteration of the process. The algorithm terminates in two cases: firstly, if an optimal solution is found subject to the fitness function and secondly, if the maximum number of iterations has been conducted.

3.4 Conclusions

The need for an integrated approach to the problem of equity portfolio management can be covered by existing multicriteria methodologies. As presented in this chapter, the MCDA methods cover a wide range of problems, as some methods are used for choice problems, others for classification problems and others for ranking problems. In addition, some methods handle continuous variables and others handle discrete variables. In fact, the MCDA sector is evolving rapidly, as new methods and improvements to existing methods are constantly being introduced. In this context, some of these methodologies focus on selecting the most attractive securities, some of them face the problem of portfolio optimization, while others focus on the problem of the comparative evaluation of these portfolios. As a result, MCDA is the most appropriate sector of operational research to address the problem of portfolio management, as it combines a wide range of continuous and discrete methodologies that can be applied to the various stages of this multiobjective decision problem.

Chapter 4
Literature Review

4.1 Introduction

In the basis of the Markowitz mean–variance methodology [112, 113], a series of new models have been developed, such as the single index models, the multi-index models, the average correlation models, the mixed models, and the utility models. Additionally, different criteria have been proposed such as geometric mean return, skewness, etc.

However, conventional theoretical approaches do not take into account the investor's specialized preferential profile and individual goals. On the other hand, MCDA offers the opportunity of incorporating the profile of the investor to the decision-making process and provides a consistent methodological basis for dealing with the problem of portfolio construction. Therefore, realistic methodologies are developed by assessing a series of significant criteria, apart from the two basic criteria of return and risk.

In this chapter, an elaborate categorized bibliographic review of the existing research activity in the fields of portfolio management and MCDA is provided. Pardalos et al. [140] have initially provided an extensive review on the use of optimization models for portfolio selection. A series of excellent literature reviews in the field of portfolio selection with multiple criteria have, also, been developed by Aouni et al. [9], Zopounidis and Doumpos [230], Patari et al. [142] and Jayasekara et al [87].

This chapter is organized as follows. Section 4.2 summarizes the major contributions made in the field of portfolio management. Section 4.3 covers the criteria that have been included to the portfolio construction process. Finally, Section 4.4 provides a categorization of the existing studies, depending on the utilized method (discrete MCDA methods, multiobjective mathematical programming, goal programming). Finally, the concluding remarks are given in Section 4.5.

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_4

4.2 Review of Major Contributions

According to Hurson and Zopounidis [80, 81], the portfolio management process consists of two phases: (a) the first phase includes the evaluation of the securities in order to select the most preferred ones, (b) the second phase includes the specification of the amount of capital to be invested in each of the securities selected in the first stage. These phases are implemented based on the classic Markowitz model. Due to the fact that both phases have multiobjective characteristics, a variety of methodologies have been developed to support the security selection process as well as the portfolio optimization. The phase of securities' evaluation has been mainly supported by discrete MCDA methods including the outranking relations theory, the multiattribute utility theory, and the preference disaggregation analysis. The highest-ranked securities are selected for portfolio synthesis purposes in the second phase of the analysis. The second phase (portfolio optimization) has been addressed as a MMP or GP problem.

The studies concerning the use of multicriteria analysis in the portfolio selection process have been classified according to their methodological framework by Zopounidis et al. [227]: (a) MMP, (b) outranking relations, (c) multiattribute utility theory, and d) preference disaggregation approach.

Additionally, Steuer et al. [188] categorizes the multicriteria portfolio analysis research into six categories: (a) overall framework, (b) portfolio ranking, (c) skewness inclusion, (d) use of alternative measures of risk, (e) decision support systems, and (f) the modeling of individual investor preferences.

In this paragraph we shortly present some of the most important pieces that contributed in the development of new methodological frameworks supporting the portfolio management process according to Xidonas et al. [219]. Lee and Chesser [104] present a GP model for portfolio selection. Rios-Garcia and Rios-Insua [150] exploit the multiattribute utility theory and multiobjective linear programming in order to construct a portfolio. Colson and de Bruyn [46] present an optimization system which allows for the construction of a multiobjective portfolio. Hurson and Zopounidis [80] proposed the multiobjective optimization system MINORA (multicriteria interactive ordinal regression analysis) for portfolio selection. Additionally, Zopounidis et al. [227] proposed the use of the ADELAIS (Aide a la decision pour systemes Lineaires multicriteres par AIde a la Structuration des preferences) multiobjective system for portfolio construction including a series of diversification constraints. Tamiz et al. [198] proposed a GP methodology for portfolio evaluation and selection.

Finally, Aouni et al. [9] present a review of the papers which have been published that apply multicriteria methods to address portfolio selection problems categorizing the portfolio management procedure into two phases: the first phase includes the security analysis and evaluation with discrete MCDA methods (AHP, PROMETHEE, etc.), while the second phase includes the portfolio construction and optimization with the techniques of goal programming, compromise programming and the weighting approach among others.

4.3 Utilized Criteria Overview

The security selection process is the first step in tackling the problem of portfolio management. Therefore, investors take into account a number of criteria that come from different sectors, such as financial criteria (profitability, liquidity, dividend yield, etc.), stock market criteria (price-to-earnings ratio, earnings per share ratio, etc.) and ESG criteria (Environmental, Social and Governance). Financial criteria include, among others, *profitability* used by Albadvi et al.[5], *return on equity* used by Doumpos et al. [58], and *return on assets* used by Xidonas et al. [213, 218]. On the other side, the most common stock market criteria that have been utilized in the security selection process are Price/Earning Ratio (Albadvi et al. [5], Marasovic and Babic [111]) and Earnings per Share Ratio (Xidonas et al. [220]). In addition to these criteria, securities evaluation is largely based on expert analysis and suggestions. With the use of all this information, the comparison of securities becomes more reliable and the chances of selecting a set of predominant shares increase.

Regarding the portfolio optimization process, many researchers have tried to replace or enrich the two basic criteria (expected return and variance) with other criteria such as systematic risk, skewness, kurtosis, and mean absolute return. As a result, models with more than two criteria have emerged, where the set of efficient solutions produces a surface, instead of the curve that is produced in the case of two objectives. It is important to note that as the number of objectives increases, so does the computational difficulty of the problem.

A series of studies for portfolio optimization have used systematic risk as an additional criterion. Systematic risk refers to the risk inherent to the entire market as a whole (unlike unsystematic risk that only refers to one security), incorporating also economic, geopolitical, and financial factors. Therefore, systematic risk is an unpredictable factor which is difficult to completely avoid. Some indicative studies that have exploited this measure include Xidonas, Askounis, and Psarras [213], Perez-Gladish, Jones, Tamiz, and Bilbao-Terol [67], Masmoudi and Abdelaziz [117] and Marasovic and Babic [111].

A large number of studies have attempted to incorporate skewness and kurtosis into the problem of portfolio optimization, in order to extend beyond the basic model that uses only the expected return and variance of returns. Skewness and kurtosis consider the extreme values of the dataset (opposed to the expected return which focuses on the average), thus being very significant metrics for judging a return distribution. Typical examples are the studies developed by Liu, Zhang, and Xu [108] and Prakash, Chang, and Pactwa [148].

Additionally, another measure of risk, which has been utilized in a number of studies, is the average absolute deviation or mean absolute deviation (MAD). The mean absolute deviation MAD of a dataset is the average of the absolute deviations from the expected value. The most indicative studies that use mean absolute deviation as an alternative way to incorporate risk are the ones developed by Xidonas, Mavrotas, Zopounidis and Psarras [220] and Tamiz, Azmi and Jones [196].

4.4 Existing Studies Categorization

In Tables 4.1–4.3 we record the studies concerning the application of multicriteria methods in the portfolio management process. This table constitutes an updated summary of the review studies developed by Zopounidis [226], Zopounidis and

Table 4.1 Discrete MCDA techniques applied in portfolio management phases.

Approach	Number of articles	Studies
PROMETHEE	5	Leao et al. (2019) [102]
		Marasovic and Babić (2011) [111]
		Albadvi et al. (2007)[5]
		Bouri et al. (2002) [28]
		Martel et al. (1991) [116]
TOPSIS	10	Fauzi et al. (2019) [61]
		Han, Zhang and Yi (2019) [71]
		Yodmun and Witayakiattilerd (2016) [225]
		Lamata et al. (2018) [101]
		Bilbao-Terol, et al. (2014) [23]
		Behzadian et al (2012) [18]
		Liu et al. (2012) [108]
		Amiri et al. (2010) [7]
		Chen and Hung (2009) [40]
		Tiryaki and Ahlatcioglu (2005) [200]
ELECTREE	7	Vezmelaia et al. (2015) [204]
		Xidonas et al. (2012) [217]
		Chen and Hung (2009) [40]
		Xidonas, Mavrotas and Psarras (2009) [218]
		Xidonas et al. (2009) [213]
		Khoury et al. (1993) [92]
		Martel, Khoury, and Bergeron (1988) [115]
AHP	13	Leao et al. (2019) [102]
		Han, Zhang and Yi (2019) [71]
		Lamata et al. (2016) [101]
		Garcia-Melon et al. (2016) [63]
		Yodmun and Witayakiattilerd (2016) [225]
		Petrillo et al. (2016) [145]
		Kiris and Ustun (2012) [93]
		Nguyen and Gordon-Brown (2012) [130]
		Marasovic and Babic (2011) [111]
		Perez-Gladish and M'Zali (2010) [144]
		Cheung and Liao (2009) [42]
		Tiryaki and Ahlatcioglu (2009) [199]
		Bouri et al. (2002) [28]

Doumpos [230], Steuer and Na [184], Spronk et al. [179], Xidonas et al. [219], Xidonas and Psarras [221], and Aouni et al. [9]. The included studies have been classified according to their methodological basis. The categorization we adopt in this chapter includes:

1. Articles that develop a discrete multicriteria methodological approach (Table 4.1)
2. Articles that develop a MMP methodological approach (Table 4.2)
3. Articles that develop a GP methodological approach (Table 4.3)

Discrete MCDA techniques are a very useful tool in the security selection phase of the problem, because they allow us to compare distinct alternatives (either with pairwise comparisons or with a utility function) and provide a ranking of the alternatives. Therefore, these methods have been utilized to find predominant securities in a series of studies. Martel et al. [116], Bouri et al. [28], and Albadvi et al. [5] incorporate the PROMETHEE method for portfolio construction. Tiryaki and Ahlatcioglu [200], Liu et al. [108] and Chen and Hung [40] present the first methodological frameworks including the TOPSIS method in order to formulate a portfolio. ELECTREE method is proposed by Xidonas et al. [213, 217] and Xidonas, Mavrotas, and Psarras [218] for portfolio selection and comparison. Finally, Bouri et al. [28], Cheung and Liao [42], and Yodmun and Witayakiattilerd (2016) [225] use the AHP methodology in order to construct a portfolio. An in-depth presentation of the contributions of multiattribute value-based and outranking relations methods is made by Marques et al.[114]. The articles that develop a MCDA methodological approach are recorded in Table 4.1:

Multiobjective mathematical programming is a tool that is mainly used in the portfolio optimization phase. Thus, many scientific studies have used this method to solve problems either with two alternatives (expected return and variance) or with more than two, combining criteria that we presented in the previous section. Some indicative studies are presented as follows. Stone [191] presents a linear programming formulation of the portfolio construction problem. Sealey [167] proposes a multiobjective model for commercial bank portfolio management. Konno and Suzuki [97] formulate a mean–variance-skewness portfolio optimization model. The purpose of Rustem [158] is to construct an efficient portfolio with nonlinear programming. Ogryczak [132, 133] proposes a linear programming model with multiple criteria. Zopounidis and Doumpos [229] propose INVESTOR, a decision support system for portfolio selection based on multiple criteria. Xidonas and Mavrotas [215] present a multiobjective portfolio optimization with non-convex policy constraints. The purpose of Mehlawat [123] is the construction of a mean-entropy model for portfolio selection. The articles that develop a MMP methodological approach are recorded in Table 4.2:

Finally, goal programming is a technique that has been widely used in recent decades, mainly to address the problem of portfolio optimization. The main advantage of this technique is that it allows a clear modeling of the problem, facilitating the investor to incorporate his personal goal-values into the objectives. The most representative studies using the goal programming technique are presented as follows. Lee and Lerro [105] present a GP methodology in order to optimize

Table 4.2 MMP techniques applied in portfolio management phases.

Approach	Number of articles	Studies
MMP	67	Khan et al. (2020) [90]
		Akbay et al. (2020) [4]
		Wei et al. (2020) [208]
		Qi (2019) [149]
		Calvo, Ivorra, and Liern (2017) [36]
		Calvo et al. (2016) [35]
		Mehlawat (2016) [123]
		Calvo et al. (2015) [34]
		Hasuikea and Katagiri (2014) [74]
		Bilbao-Terol et al. (2013) citeBilbao
		Garcia, Guijarro, and Moya (2013) [64]
		Xidonas et al. (2012) [220]
		Calvo, Ivorra, and Liern (2012) [33]
		Xidonas and Mavrotas (2012) [215]
		Moon and Yao (2011) [125]
		Ammar (2009) [8]
		Xidonas et al. (2009) [214]
		Sevastjanov and Dymova (2009) [168]
		Branke et al. (2008) [29]
		Steuer et al. (2008a, b) [188] [189]
		Mavrotas et al. (2008) [121]
		Xidonas et al. (2008) [216]
		Steuer et al. (2007) [187]
		Steuer et al. (2006a) [186]
		Roman et al. (2007) [152]
		Ahmed and El-Alem (2005) [3]
		Steuer et al. (2005) [185]
		Ehrgott et al. (2004) [59]
		Schlottmann and Seese (2004) [162]
		Prakash et al. (2003) [148]
		Ogryczak (2002) [132]
		Leung et al. (2001) [106]
		Ogryczak (2000) [133]
		Schniederjans and Schniederjans (2000) [164]
		Zopounidis and Doumpos (2000) [229]
		Bertsimas et al. (1999) [21]
		Mansini and Speranza (1999) [110]
		Shing and Nagasawa (1999) [171]
		Zopounidis et al. (1998) [227]
		Rustem (1998) [158]
		Chunhachinda et al. (1997) [43]
		Hurson and Zopounidis (1997) [80]

(continued)

Table 4.2 (continued)

Approach	Number of articles	Studies
		Coffin and Taylor (1996) [45]
		Skulimowski (1996) [175]
		Speranza (1996) [177]
		Hurson and Zopounidis (1996) [81]
		Chen (1995) [41]
		Konno and Suzuki (1995) [97]
		L'Hoir and Teghem (1995) [100]
		Speranza (1994) [178]
		Konno et al. (1993) [96]
		Weber and Current (1993) [207]
		Speranza (1993) [176]
		Rys and Ziemba (1991) [159]
		Skocz et al. (1989) [174]
		Kobayashi et al. (1987) [94]
		Nakayama et al. (1983) [129]
		Rios-Garcia and Rios-Insua (1983) [150]
		Muhlemann and Lockett (1980) [127]
		Wilhelm (1980) [210]
		Muhlemann et al. (1978) [128]
		Sealey (1977) [167]
		Shapiro (1976) [169]
		Caplin and Kornbluth (1975) [37]
		Stone and Reback (1975) [192]
		Steuer (1974) [190]
		Stone (1973) [191]

the portfolio selection of mutual funds. El Sheshai [170] uses integer GP for cost volume profit analysis. Kumar et al. [99] propose a GP model for portfolio selection for dual-purpose funds. Colson and Bruyn [46] incorporate GP in an integrated multiobjective portfolio management system. Powell and Premachandra [147] use a nonlinear GP methodology. Tamiz et al. [198] propose the use of a GP model for portfolio evaluation and selection. The articles that develop a GP methodological approach can be shown in Table 4.3:

4.5 Conclusions

In this chapter we discussed the contribution of MCDA in portfolio management problems. MCDA methods, in general, have helped researchers to broaden the limited and narrow framework of single-objective optimization. The main advan-

Table 4.3 GP techniques applied in portfolio management phases.

Approach	Number of articles	Studies
GP	56	Tamiz and Azmi (2019) [195]
		Bagheri (2019) [13]
		Gupta et al. (2019) [69]
		Masri (2018) [118]
		Masmoudi and Abdelaziz (2017) [117]
		Messaoudi et al. (2017) [124]
		Bilbao-Terol et al. (2016a, 2016b) [25, 26]
		Mehlawat (2016) [123]
		Kocadagli and Keskin (2015) [95]
		Abdelaziz and Masmoudi (2014) [2]
		Trenado et al. (2014) [201]
		Briec et al. (2013) [32]
		Ghahtarani and Najafi (2013) [65]
		Tamiz et al. (2013) [196]
		Gupta et al. (2013) [70]
		Ballestero et al. (2012) [15]
		Ballestero and Garcia-Bernabeu (2012) [16]
		Bilbao-Terol et al. (2012) [22]
		Bravo et al. (2010) [31]
		Abdelaziz, El Fayedh and Rao (2009) [1]
		Ballestero et al. (2009) [17]
		Davis, Kat and Lu (2009) [51]
		Wu, Chou, Yang and Ong (2007) [211]
		Perez-Gladish et al. (2007) [67]
		Bilbao-Terol, et al. (2006) [24]
		Sugrue et al. (2006) [194]
		Pendaraki et al. (2005) [143]
		Prakash et al. (2003) [148]
		Arenas Parra et al. (2001) [141]
		Ballestero (2001) [14]
		Zopounidis and Doumpos (2000) [229]
		Doumpos et al. (1999) [57]
		Powell and Premachandra (1998) [147]
		Cooper et al. (1997) [47]
		Dominiak (1997) [55]
		Chunhachinda et al. (1997) [43]
		Tamiz et al. (1997) [197]
		Tamiz et al. (1996) [198]
		Khorramshahgol and Okoruwa (1994) [91]
		Schniederjans et al. (1993) [166]
		Vermeulen et al. (1993) [203]

(Continued)

Table 4.3 (Continued)

Approach	Number of articles	Studies
		Colson and Bruyn (1989) [46]
		Alexander and Resnik (1985) [6]
		Spronk and Zambruno (1985) [181]
		Schniederjans (1984)[165]
		Spronk and Veeneklaas (1983) [180]
		Spronk and Zambruno (1981) [182]
		Harrington and Fischer (1980) [72]
		Lee and Chesser (1980) [104]
		Kumar and Philippatos (1979) [98]
		Kumar et al. (1978) [99]
		Taylor and Keown (1978) [20]
		Booth and Dash (1977) [27]
		El Sheshai et al. (1977) [170]
		Orne et al. (1975) [137]
		Lee and Lerro (1973) [105]

tages of the MCDM, as stated by Zopounidis [226] and Zopounidis and Doumpos [230], are: (a) the possibility of structuring complex evaluation problems, (b) the introduction of both quantitative and qualitative criteria in the evaluation process, (c) the transparency in the evaluation, allowing good argumentation in financial decisions, and (d) the introduction of sophisticated, flexible, and realistic scientific methods in the financial decision-making process. In this context, the mean–variance methodological framework has been enriched with a series of criteria, which aspire to incorporate all aspects of the problem and the MCDA field provides all the necessary methods for these criteria to be evaluated on a strong methodological basis. In conclusion, MCDA facilitates the portfolio management procedure, enabling the decision-maker to participate actively in the financial decision-making process, thus leading to more flexible and realistic methodological frameworks.

Chapter 5
The Proposed Methodology

5.1 Introduction

As already mentioned, the purpose of this book is the development of an integrated methodological framework for portfolio management. The process of portfolio management is a very complex problem, as it consists of two phases which demand a series of decisions. The first phase focuses on portfolio selection, i.e. the selection of the strongest investment opportunities. The second phase includes portfolio optimization, the determination of the most efficient allocation of the available capital to the selected securities in order to maximize return.

In this chapter, we present the proposed methodology. The following analysis is based on Xidonas et al. scientific work presented on Xidonas thesis project [212, 220], as well as the book Multicriteria Portfolio Management [217]. The ultimate goal is the effective management of security portfolios, which constitute one of the riskiest market investments. The proposed methodology aspires to combine existing knowledge with a set of theoretical and practical innovations. In the first phase, four multicriteria decision-making methods are deployed in order to rank the available securities and detect the best investment opportunities. After the security selection process, the most important financial statistical indexes are calculated based on historical data and, subsequently, a series of portfolio optimization models are proposed. The basis of these models is the typical mean–variance model, which remains the basic portfolio optimization method for more than 60 years.

This chapter has the following structure: In Section 5.2 an overview of the proposed methodology is presented. The phases of multicriteria portfolio selection and multiobjective portfolio optimization are explicitly analyzed in sections 5.3 and 5.4, respectively.

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_5

5.2 Methodology Overview

In the current section an overview of the proposed methodological framework will be provided, as an introduction to the various methods which are discussed in the following sections. The methodological framework consists of two main phases: (a) portfolio selection and (b) portfolio optimization as indicated in Figure 5.1.

The aim of the following overview is to unify all the individual methodologies in a structured and compact framework. As shown in Figure 5.1, the two main phases are not independent from each other. On the contrary, they are interdependent as there is a strong connection and communication between them. Additionally, the whole methodology should be applied in communication with the decision-maker, as it is necessary that the decision-maker interacts with the model importing his preferences during the process. An extensive diagram of the proposed methodological framework is presented in Figure 5.2.

5.2.1 Phase I: Multicriteria Portfolio Selection

The first phase involves the portfolio selection problem, that is the construction of a portfolio of securities which are the best investment opportunities. The decision-maker must select the industrial sector and the stock exchange that he wants to get involved with, resulting in a pool of securities which constitute the problem alternatives. The problem of security selection is cured with multicriteria decision analysis (MCDA). More specifically, four ranking MCDA methods are applied to the alternatives, under a variety of financial indexes which serve as the criteria of the multiobjective optimization problem. Each method provides a ranking of the securities and finally the cumulative ranking of the securities can be calculated as the weighted average of the four individual rankings. After the whole process, the

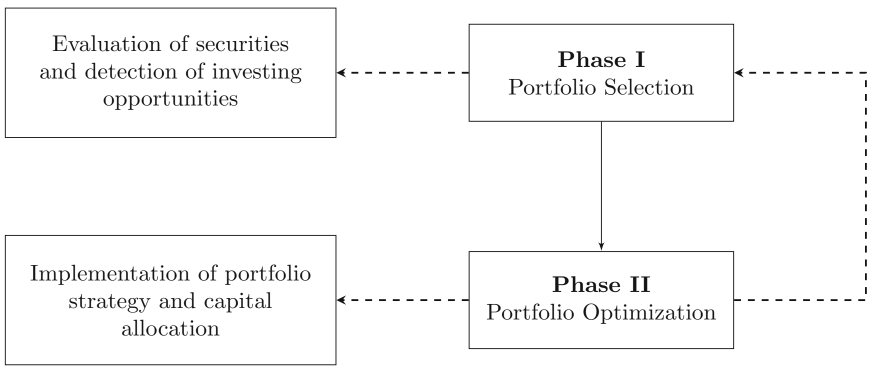


Fig. 5.1 Portfolio management methodology phases

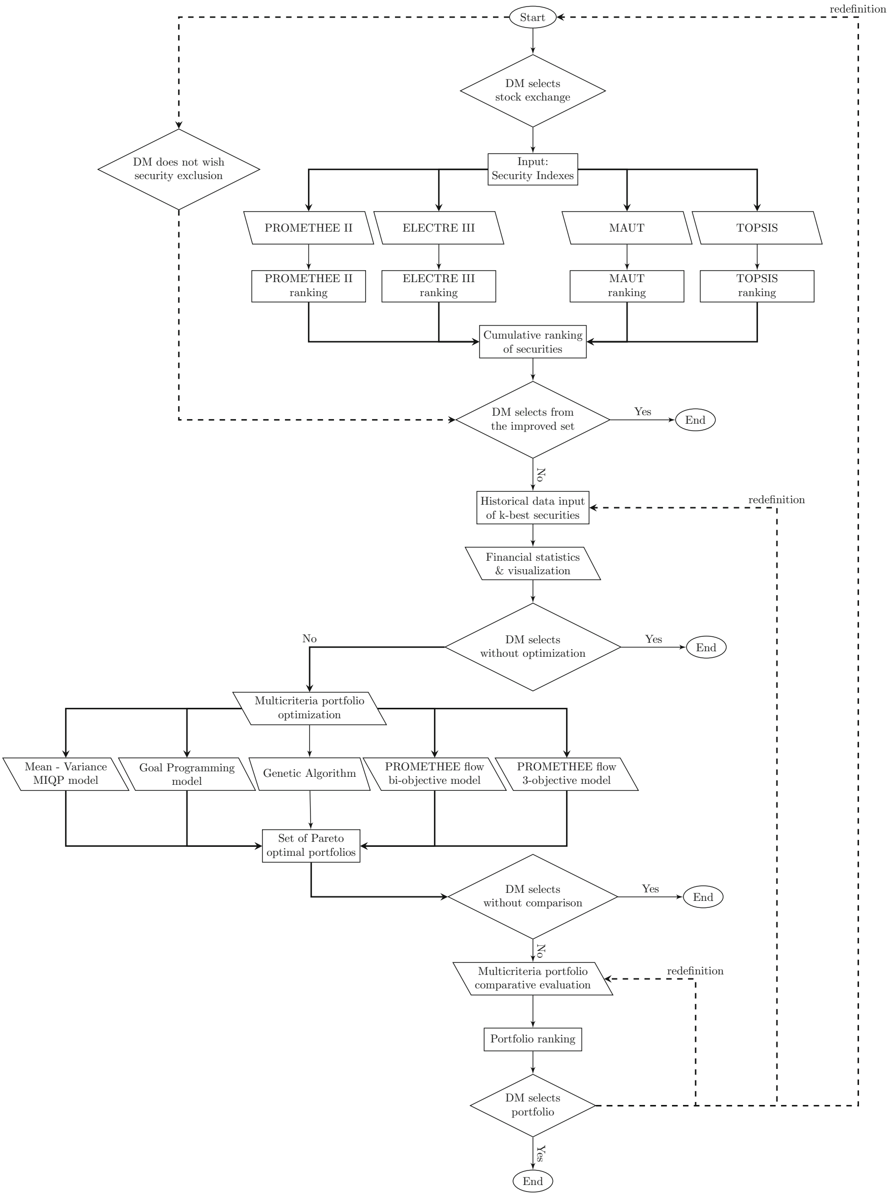


Fig. 5.2 Methodological framework extensive presentation

decision-maker can either construct a portfolio with the $k-$best ranked securities, or redefine the problem in case that the result is unsatisfying.

5.2.2 Phase II: Multiobjective Portfolio Optimization

The second phase involves the problem of portfolio optimization. The original optimization problem was a bi-criteria problem where the expected return is maximized and the risk is minimized (Markowitz 1952) [112]. The methodological framework proposes a series of models to approach the problem. Firstly, a bi-objective integer programming model is formulated based on the mean–variance approach, where additional integer constraints are imposed in order to control the weighting factor of each security. Secondly, another bi-objective optimization approach is introduced where the net flow of PROMETHEE method should be maximized and portfolio beta index should be minimized. Additionally, a goal programming methodology is introduced. Finally, an implementation of a genetic algorithm for portfolio optimization is presented.

5.3 Phase I: Multicriteria Portfolio Selection

Multicriteria decision analysis (MCDA) methods are widely used for the study of a wide variety of financial problems. More specifically, the problem of security selection involves all the features of MCDA, such as alternatives, evaluation criteria, and objective functions, rendering it one of the most suitable approaches. In this phase of the process, the decision-maker must evaluate and select the securities that are available as investment opportunities. This step is necessary because of the vast amount of securities traded in international stock markets. Consequently, the portfolio should consist of a limited number of those securities, excluding securities which have undesirable characteristics.

Before presenting a detailed description of the problem, it is necessary to emphasize the following characteristics of the methodology: Firstly, the process of security evaluation is based on specific financial indexes (problem criteria), after an extensive study of the existing literature. Secondly, the companies should be categorized into predefined classes before the evaluation is applied, depending on their activity and the industrial sector they belong to. The necessity of this step derives from the fact that the comparison of financial indexes among companies of different industrial sectors would be a contentious assumption.

The methodological framework of the first phase is depicted in Figure 5.3. In the following paragraphs, a detailed description of the portfolio selection problem is given:

Problem Definition

Let $A = \{a_1, \ldots, a_n\}$ be a set of n alternatives and let $F = \{f_1, \ldots, f_q\}$ be a consistent family of q criteria. Without loss of generality, we assume that the above criteria should be maximized. Therefore, let us consider the following multicriteria problem:

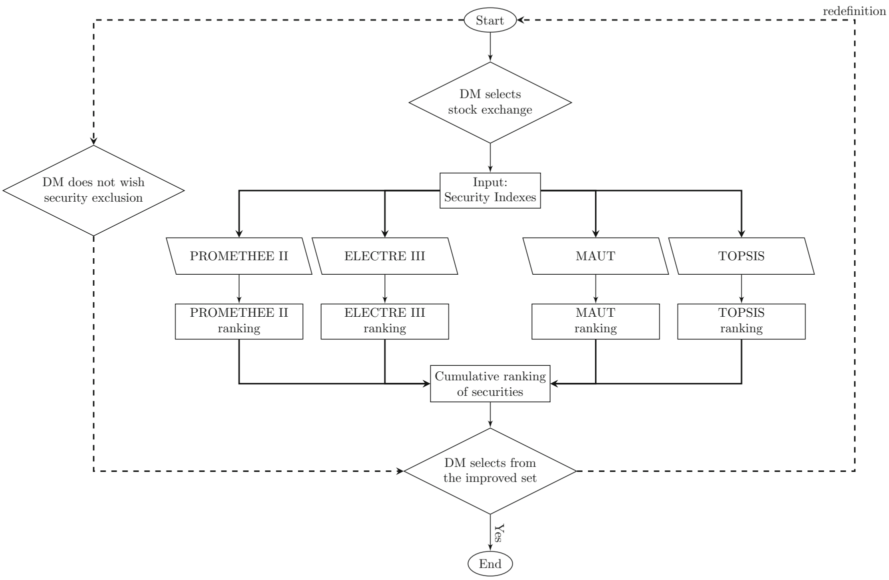


Fig. 5.3 Methodological framework for Phase I

Table 5.1 MCDA problem evaluation table

	$f_1(\cdot)$	$f_2(\cdot)$	$\cdots$	$f_j(\cdot)$	$\cdots$	$f_q(\cdot)$
a_1	$f_1(a_1)$	$f_2(a_1)$	$\cdots$	$f_j(a_1)$	$\cdots$	$f_q(a_1)$
a_2	$f_1(a_2)$	$f_2(a_2)$	$\cdots$	$f_j(a_2)$	$\cdots$	$f_q(a_2)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
a_i	$f_1(a_i)$	$f_2(a_i)$	$\cdots$	$f_j(a_i)$	$\cdots$	$f_q(a_i)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
a_n	$f_1(a_n)$	$f_2(a_n)$	$\cdots$	$f_j(a_n)$	$\cdots$	$f_q(a_n)$

$$\max\{f_1(a), f_2(a), \ldots, f_q(a)|a \in A\} \tag{5.1}$$

The input of the above problem is imported in a 2-dimensional table, containing $n \times q$ evaluations, which is called evaluation matrix. Each row corresponds to an alternative action and each column corresponds to an evaluation criterion. The element of the i_{th} row and the k_{th} column describes the performance of alternative a_i in criterion f_k. The evaluation matrix of a multicriteria decision problem is presented in Table 5.1.

Problem Alternatives

As already mentioned, the set of alternatives $A = \{a_1, \ldots, a_n\}$ consists of the securities of a specific stock exchange and a specific industrial sector. In the beginning of the process the decision-maker selects the industrial sector, as well as the stock exchange. These two components constitute the environment of the

Table 5.2 Classification of industrial sectors

Class	Industrial sector
A	Basic materials
B	Capital goods
C	Consumer cyclical
D	Consumer non-cyclical
E	Energy
F	Financial
G	Healthcare
H	Services
I	Technology
J	Transportation
K	Utilities

Table 5.3 Problem criteria

	Criteria	Utility	Units
A	Price-to-Earnings Ratio	Minimization	Percentage
B	Earnings per share	Maximization	Percentage
C	Revenue	Maximization	Dollars
D	Beta	Minimization	Fraction
E	Dividend Yield	Maximization	Percentage
F	Monthly technical recommendation	Maximization	Rank
G	Year-to-date performance	Maximization	Percentage
H	1-year performance	Maximization	Percentage

study. The set of securities have been classified in 11 classes representing that the main industrial sectors are presented in Table 5.2.

Problem Criteria
The evaluation process of securities is based on a set of suitable financial criteria, which depend on the accounting and economic plans of the companies, as well as on experts' analysis (Table 5.3).

Offset and Thresholds Assignment
The determination of the offset of the criteria is a matter of great significance for the efficiency of multicriteria techniques. The main methodologies for offset determination are: (a) the direct weighting system (Hokkanen - Salminen, [76]), (b) the Mousseau system (Mousseau, [126]), (c) the pack of cards technique (Simos, [172]), and (d) the resistance to change grid method (Rogers and Bruen, [151]). The process of offset allocation must be developed with the assistance of the decision-maker, because his profile and his preferences among the significance of conflicting criteria must be considered during the decision support model.

Some of the MCDA methods that are used in the proposed methodology include some additional thresholds. In the proposed techniques there are three types of thresholds: (a) *preference threshold*, (b) *indifference threshold* and (c) *veto threshold*. Preference threshold implies that an alternative is totally preferable to another. Indifference threshold signifies that two alternatives are almost equally preferred. Finally, veto threshold represents the threshold that renders a dominated alternative eliminated from the selection process. Threshold determination is a quite complicated process which should be executed in communication with financial experts.

5.3.1 ELECTRE III

The ELECTRE family in MCDA problems is based on the concept of outranking relationship. An alternative a_1 outranks a_2 if and only if there is sufficient evidence to believe that a_1 is better than a_2 or at least a_1 is as good as a_2. More specifically, ELECTRE III method is used for ranking problems, using a structured procedure to calculate the outranking relationship between each pair of alternatives. It includes a preference threshold, an indifference threshold, and a veto threshold.

Let $q(f_i)$ and $p(f_i)$ represent the indifference and preference thresholds for each criterion f_i, $i = 1, \ldots, q$, respectively, and let P denote a strong preference, Q denote a weak preference, and I denote indifference between a_1 and a_2 for criterion k. If $f_k(a_i) \geq f_k(a_j)$, then

$$f_k(a_i) > f_k(a_j) + p(f_k) \Leftrightarrow a_1 P a_2$$

$$f_k(a_j) + q(f_k) < f_k(a_i) < f_k(a_j) + p(f_k) \Leftrightarrow a_1 Q a_2$$

$$f_k(a_j) < f_k(a_i) < f_k(a_j) + q(f_k) \Leftrightarrow a_1 I a_2$$

The algorithm of the ELECTRE III method is presented below:

Step 1: Outranking Degree The outranking degree $C_k(a_i, a_j)$, $(0 \leq C_k(a_i, a_j) \leq 1)$ of the alternative a_i and the alternative a_j for criterion f_k is calculated according to the preference definitions (linear interpolation):

$$C_k(a_i, a_j) = \begin{cases} 0 & \text{if} \quad f_k(a_j) - f_k(a_i) > p(f_k) \\ 1 & \text{if} \quad f_k(a_j) - f_k(a_i) \leq q(f_k) \\ \dfrac{p(f_k) + f_k(a_i) - f_k(a_j)}{p(f_k) - q(f_k)} & \text{otherwise} \end{cases} \tag{5.2}$$

Step 2: Concordance Index The concordance index $C(a_i, a_j)$ is computed for each pair of alternatives a_i, a_j, as follows:

Algorithm 2: ELECTRE III Algorithm

input: n (alternatives), q (criteria), $f_i(a_j)$ (evaluation table);
for *all pairs of alternatives* $a_i,\ a_j,\ i,\ j \in \{1, \ldots, n\}$ **do**
  for *all criteria* $f_k,\ k \in \{1, \ldots, q\}$ **do**
    compute outranking degree $C_k(a_i, a_j)$
  end
end
for *all pairs of alternatives* $a_i,\ a_j,\ i,\ j \in \{1, \ldots, n\}$ **do**
  compute concordance index $C(a_i, a_j)$
end
for *all pairs of alternatives* $a_i,\ a_j,\ i,\ j \in \{1, \ldots, n\}$ **do**
  for *all criteria* $f_k,\ k \in \{1, \ldots, q\}$ **do**
    compute discordance index $D_k(a_i, a_j)$
  end
end
for *all pairs of alternatives* $a_i,\ a_j,\ i,\ j \in \{1, \ldots, n\}$ **do**
  compute degree of outranking relationship $S(a_i, a_j)$
end
for *all alternatives* $a_i,\ i \in \{1, \ldots, n\}$ **do**
  compute concordance credibility index $\phi^+(a_i)$
  compute discordance credibility index $\phi^-(a_i)$
  compute net credibility index $\phi(a_i)$
end
$FinalRanking =$ **SortDesc**(ϕ)
Result: FinalRanking

$$C(a_i, a_j) = \frac{\sum_{k=1}^{q} w_k C_k(a_i, a_j)}{\sum_{k=1}^{q} w_k} \tag{5.3}$$

Step 3: Discordance Index Let $v(f_k)$ represent the veto threshold for criterion f_k. The veto threshold rejects the possibility of $a_i S a_j$ if, for any criterion f_k, the relationship $f_k(a_j) > f_k(a_i) + v(f_k)$ is satisfied. The discordance index $D(a_i, a_j)$, $(0 \leq D_k(a_i, a_j) \leq 1)$ for each criterion is defined as follows:

$$D_k(a_i, a_j) = \begin{cases} 0 & \text{if} \quad f_k(a_j) - f_k(a_i) \leq p(f_k) \\ 1 & \text{if} \quad f_k(a_j) - f_k(a_i) > v(f_k) \\ \dfrac{f_k(a_j) - f_k(a_i) - p(f_k)}{v(f_k) - p(f_k)} & \text{otherwise} \end{cases} \tag{5.4}$$

Step 4: Reliability Index Let $J(a_i, a_j)$ represent the set of criteria for which $D_k(a_i, a_j) > C(a_i, a_j)$. The reliability index $S(a_i, a_j)$ is

$$S(a_i, a_j) = \begin{cases} C(a_i, a_j) & \text{if} \quad D_k(a_i, a_j) \leq C(a_i, a_j) \; \forall k \in J \\ C(a_i, a_j) \times \prod_{k \in J(a_i, a_j)} \dfrac{1 - D_k(a_i, a_j)}{1 - C(a_i, a_j)} & \text{otherwise} \end{cases} \tag{5.5}$$

Step 5: Concordance and Discordance Credibility Degrees The concordance credibility degree $\phi^+(a_i)$ is an indicator that measures how an alternative a_i dominates all the other alternatives [73]. The definition of concordance credibility degree is

$$\phi^+(a_i) = \sum_{x \in A} S(a_i, x) \tag{5.6}$$

The discordance credibility degree $\phi^-(a_i)$ is an indicator that measures how an alternative a_i is dominated by all the other alternatives. The definition of discordance credibility degree is

$$\phi^-(a_i) = \sum_{x \in A} S(x, a_i) \tag{5.7}$$

Step 6: Net Credibility Degree Finally, the net credibility degree $\phi(a_i)$ is an indicator of the value of the alternative a_i. A higher net credibility degree implies a better alternative. The definition of the net credibility degree for an alternative a_i is

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i) \tag{5.8}$$

The ELECTRE III final ranking is obtained by ordering the alternatives according to the decreasing values of the net flow scores.

Numerical Example Let us consider the case of four securities, which are the alternatives of the problem and three criteria. In Table 5.4 we show the input table, the thresholds, and the offsets of the problem. We can assume that all criteria are maximization criteria. The presented example is also implemented in Python programming language and it is accessible in the following link: https://github.com/epu-ntua/Multicriteria-Portfolio-Construction-with-Python.

The ELECTRE III concordance table as calculated by steps 1 and 2 is shown in Table 5.5. Below, we show the calculations extensively for Securities 1 and 3.

$C_1(a_1, a_3) = C_2(a_1, a_3) = C_3(a_1, a_3) = 1$, because $f_k(a_3) - f_k(a_1) \leq q(f_k), \quad \forall k$. Therefore, $C_1(a_1, a_3) = 0.4 \times 1 + 0.1 \times 1 + 0.5 \times 1 = 1$

Criterion 1:

$$C_1(a_3, a_1) = (10 + 12 - 20)/(10 - 5) = 0.4$$

Table 5.4 Numerical Example Input Table

Alternatives	Criterion 1	Criterion 2	Criterion 3
Sec1	20	5	20
Sec2	15	3	5
Sec3	12	2	10
Sec4	10	4	35
Weight	0.4	0.1	0.5
Preference Criterion	10	2	15
Indifference Criterion	5	1	5
Veto Criterion	20	5	30

Table 5.5 ELECTRE III Concordance Table

	Sec1	Sec2	Sec3	Sec4
Sec1	0	1	1	0.5
Sec2	0.4	0	1	0.5
Sec3	0.41	1	0	0.4
Sec4	0.6	1	1	0

Table 5.6 ELECTRE III Discordance Table for criterion 3

	Sec1	Sec2	Sec3	Sec4
Sec1	0	0	0	0
Sec2	0	0	0	1
Sec3	0	0	0	0.67
Sec4	0	0	0	0

Criterion 2:

$$C_2(a_3, a_1) = 0, \quad \text{because} \quad 5 - 2 = 3 > 2 = p_2$$

Criterion 3:

$$C_3(a_3, a_1) = (15 + 10 - 20)/(15 - 5) = 0.5$$

Therefore:

$$C(a_3, a_1) = 0.4 \times 0.4 + 0.1 \times 0 + 0.5 \times 0.5 = 0.41$$

The ELECTRE III discordance table for Criterion 3 as calculated by step 3 is shown in Table 5.6. Accordingly, we can calculate the discordance tables for criteria 1 and 2. As an example, we show the calculation for securities 3 and 4, for criterion 3.

$$D_3(a_3, a_4) = (35 - 10 - 15)/(30 - 15) = 10/15 \approx 0.67$$

The ELECTRE III reliability index as calculated by step 4 is shown in Table 5.7.

$$S(a_3, a_4) = 0.4 \times (1 - 0.67)/(1 - 0.4) = 0.4 \times 0.33/0.6 = 0.22$$

Table 5.7 ELECTRE III Reliability Index Table

	Sec1	Sec2	Sec3	Sec4
Sec1	1	1	1	0.5
Sec2	0.4	1	1	0
Sec3	0.41	1	1	0.22
Sec4	0.6	1	1	1

Table 5.8 ELECTRE III Concordance, Discordance, and Net Degrees

	Concordance Flow	Discordance Flow	Net Flow
Sec1	3.5	2.41	1.09
Sec2	2.4	4	−1.6
Sec3	2.63	4	−1.37
Sec4	3.6	1.72	1.88

Finally, the ELECTRE III concordance, discordance, and net credibility degrees (or flows) as calculated by steps 5, 6, and 7 are shown in Table 5.8.

$$\phi^{+}(a_1) = 1 + 1 + 1 + 0.5 = 3.5$$

$$\phi^{-}(a_1) = 1 + 0.4 + 0.41 + 0.6 = 2.41$$

$$\phi(a_1) = \phi^{+}(a_1) - \phi^{-}(a_1) = 3.5 - 2.41 = 1.09$$

5.3.2 *PROMETHEE II*

The PROMETHEE I and PROMETHEE II methods were introduced by J.P. Brans [30] at a conference at the Université Laval, Québec, Canada (L'Ingéniérie de la Décision. Elaboration d'instruments d'Aide à la Décision) and have been extensively applied since then in fields such as business, healthcare, and education. The acronym PROMETHEE stands for Preference Ranking Organization METHod for Enrichment of Evaluations. PROMETHEE I is a partial ranking of the actions, as it is based on the positive and negative flows. It includes preferences, indifferences, and incomparabilities. On the contrary, PROMETHEE II is a complete ranking of the actions, as it is based on the multicriteria net flow. It includes a preference threshold and an indifference threshold which will be explained in the following paragraphs.

The algorithm of the PROMETHEE II method is presented below:

Step 1: Pairwise comparisons Firstly, pairwise comparisons are made between all the alternatives for each criterion. $d_k(a_i, a_j)$ is the difference between the evaluations of alternatives a_i and a_j for criterion f_k:

$$d_k(a_i, a_j) = f_k(a_i) - f_k(a_j) \tag{5.9}$$

Algorithm 3: PROMETHEE II Algorithm

input: n (alternatives), q (criteria), $f_i(a_j)$ (evaluation table), P_i (preference function);
for *all criteria* f_k, $k \in \{1, \ldots, q\}$ **do**
  $d_k(a_i, a_j) = f_k(a_i) - f_k(a_j)$
end
for *all pairs of alternatives* a_i, a_j, $i, j \in \{1, \ldots, n\}$ **do**
  $\pi_k(a_i, a_j) = P_k[d_k(a_i, a_j)]$
end
for *all criteria* f_k, $k \in \{1, \ldots, q\}$ **do**
  $\pi(a_i, a_j) = \mathbf{sum}(\pi_k(a_i, a_j))$
end
for *all pairs of alternatives* a_i, a_j, $i, j \in \{1, \ldots, n\}$ **do**
  $\phi^+(a_i) = \mathbf{sum}(\pi(a_i, a_j))$
  $\phi^-(a_i) = \mathbf{sum}(\pi(a_j, a_i))$
end
for *all alternatives* a_i, $i \in \{1, \ldots, n\}$ **do**
  $\phi(a_i) = \phi^+(a_i) - \phi^-(a_i)$
end
$FinalRanking = \mathbf{SortDesc}(\phi)$
Result: FinalRanking

Step 2: Preference degree The differences calculated in step 1 are translated to preference degrees, according to the selected criterion, as follows:

$$\pi_k(a_i, a_j) = P_k[d_k(a_i, a_j)] \tag{5.10}$$

where $P_k : R \rightarrow [0, 1]$ is a positive non-decreasing preference function such that $P_j(0) = 0$. Six different types of preference functions are proposed by PROMETHEE method. These functions are presented at the end of the section.

Step 3: Multicriteria preference degree The pairwise comparison of the alternatives is completed computing the multicriteria preference degree of each pair, as follows:

$$\pi(a_i, a_j) = \sum_{i=1}^{q} \pi_k(a_i, a_j) \cdot w_k \tag{5.11}$$

where w_k represents the weight of criterion f_k, assuming that $w_k \geq 0$ and $\sum_{k=1}^{q} w_k = 1$.

Step 4: Multicriteria preference flows Let us define the two following outranking flows:
The positive outranking flow expresses how an alternative is outranking all the others, demonstrating its outranking character. A higher positive outranking flow implies a better alternative.

$$\phi^{+}(a) = \frac{1}{n-1} \sum_{x \in A} \pi(a, x) \tag{5.12}$$

The negative outranking flow expresses how an alternative is outranked by all the others, demonstrating its outranked character. A lower positive outranking flow implies a better alternative.

$$\phi^{-}(a) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a) \tag{5.13}$$

An ideal alternative would have a positive outranking flow equal to 1 and a negative outranking flow equal to 0. The positive and negative outranking flows are aggregated into the net preference flow:

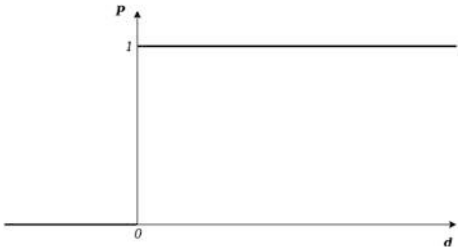
$$\phi(a) = \phi^{+}(a) - \phi^{-}(a) \tag{5.14}$$

The PROMETHEE II final ranking is obtained by ordering the alternatives according to the decreasing values of the net flow scores.

PROMETHEE Preference Functions Let d_j be the difference of two alternatives a_i, a_j and let q_j, p_j be the indifference and preference thresholds. These parameters are explained, as follows: when the difference d_j is smaller than the indifference threshold, then it is considered as negligible, therefore the preference degree becomes equal to zero. On the contrary, if the difference is larger than the preference threshold, then it is considered significant, therefore the preference degree is equal to one. Otherwise, if the difference is between the two thresholds, the preference degree is computed using a linear interpolation. The six criteria are presented below:

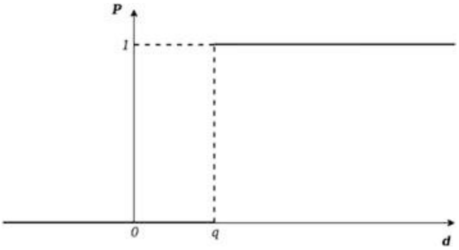
Usual Criterion

$$P_j(d_j) = \begin{cases} 0 & \text{if } d_j \leq 0 \\ 1 & \text{if } d_j > 0 \end{cases}$$



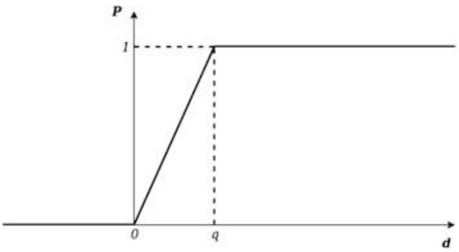
U-Shape Criterion

$$P_j(d_j) = \begin{cases} 0 \text{ if } |d_j| \le q_j \\ \\ 1 \text{ if } |d_j| > q_j \end{cases}$$



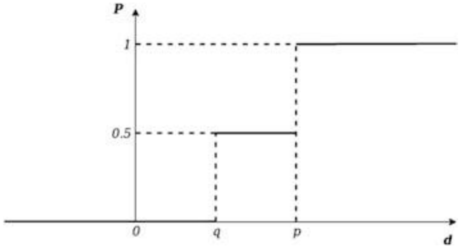
V-Shape Criterion

$$P_j(d_j) = \begin{cases} \dfrac{|d_j|}{p_j} \text{ if } |d_j| \le p_j \\ \\ 1 \quad\quad \text{if } |d_j| > p_j \end{cases}$$



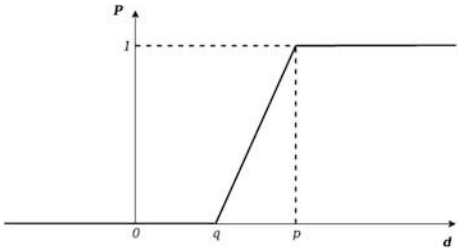
Level Criterion

$$P_j(d_j) = \begin{cases} 0 \text{ if } |d_j| \le q_j \\ \\ \frac{1}{2} \text{ if } q_j < |d_j| \le p_j \\ \\ 1 \text{ if } |d_j| > p_j \end{cases}$$



Linear Criterion

$$P_j(d_j) = \begin{cases} 0 & \text{if } |d_j| \leq q_j \\ \dfrac{|d_j| - q_j}{p_j - q_j} & \text{if } q_j < |d_j| \leq p_j \\ 1 & \text{if } |d_j| > p_j \end{cases}$$



Gaussian Criterion

$$P_j(d_j) = 1 - e^{-\frac{d_j^2}{2s_j^2}}$$

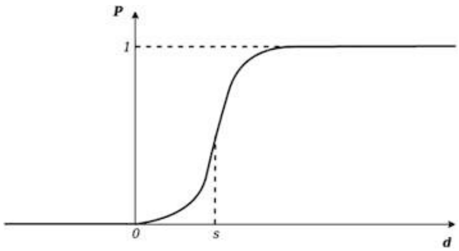


Table 5.9 PROMETHEE II Multicriteria Preference Degree

	Sec1	Sec2	Sec3	Sec4
Sec1	0	0.6	0.59	0.4
Sec2	0	0	0	0
Sec3	0	0	0	0
Sec4	0.5	0.5	0.6	0

Table 5.10 PROMETHEE II Positive, Negative, and Net Flows

	Positive Flow	Negative Flow	Net Flow
Sec1	0.53	0.16	0.37
Sec2	0	0.36	−0.36
Sec3	0	0.39	−0.39
Sec4	0.53	0.13	0.40

Numerical Example Let us consider the example of Table 5.4, assuming that we use the Linear Criterion. The multicriteria preference degree after the pairwise comparisons as calculated by steps 1, 2, and 3 is shown in Table 5.9. Below, we show the calculations extensively for Securities 1 and 3.

Criterion 1:

$$\pi_1(a_1, a_3) = (8 - 5)/(10 - 5) = 0.6$$

Criterion 2:

$$\pi_2(a_1, a_3) = 1, \quad \text{because} \quad 5 - 2 = 3 > 2 = p_2$$

Criterion 3:

$$\pi_3(a_1, a_3) = (10 - 5)/(15 - 5) = 0.5$$

Therefore:

$$\pi(a_1, a_3) = 0.4 \times 0.6 + 0.1 \times 1 + 0.5 \times 0.5 = 0.59$$

On the other side, $\pi(a_3, a_1) = 0$, because $f_k(a_3) < f_k(a_1), \quad \forall k.$

Finally, the PROMETHEE II positive, negative, and net flows as calculated by step 4 are shown in Table 5.10.

$$\phi^+(a_1) = (0.6 + 0.59 + 0.4)/3 = 1.59/3 = 0.53$$

$$\phi^-(a_1) = (0 + 0 + 0.5)/3 = 0.5/3 \approx 0.16$$

$$\phi(a_1) = \phi^+(a_1) - \phi^-(a_1) = 0.53 - 0.16 = 0.37$$

5.3.3 MAUT

Multiattribute Utility Theory (MAUT) (Keeney and Raiffa, 1993) [89] is a structured methodology which was originally designed in order to handle the trade-offs among multiple objective functions. MAUT belongs to the family of multicriteria utility theory, it can be considered as an additive value function and it has the advantage that it is adaptable to the profile of the DM, as it can describe optimistic and pessimistic behaviors.

The algorithm of the MAUT method is presented below:

Algorithm 4: MAUT Algorithm

input: n (alternatives), q (criteria), w_i (weights), $f_i(a_j)$ (evaluation table);
for *all alternatives* a_i, $i \in \{1, \ldots, n\}$ **do**
  for *all criteria* f_k, $k \in \{1, \ldots, q\}$ **do**
    compute normalized decision matrix $x_k(a_i)$
  end
end
for *all alternatives* a_i, $i \in \{1, \ldots, n\}$ **do**
  for *all criteria* f_k, $k \in \{1, \ldots, q\}$ **do**
    compute integrated DM's attitude $u_k(a_i)$
  end
end
for *all alternatives* a_i, $i \in \{1, \ldots, n\}$ **do**
  compute utility $U(i) = \mathbf{sum}(w_k u_k(a_i))$
end
$FinalRanking = \mathbf{SortDesc}(U)$
Result: FinalRanking

Step 1: Normalized Decision Matrix Let $f_k(a_{min})$, $f_k(a_{min})$ represent the minimum and maximum value for criterion k. The evaluation table is normalized, as follows:

For maximization criteria:

$$x_k(a_i) = \frac{f_k(a_i) - f_k(a_{min})}{f_k(a_{max}) - f_k(a_{min})} \tag{5.15}$$

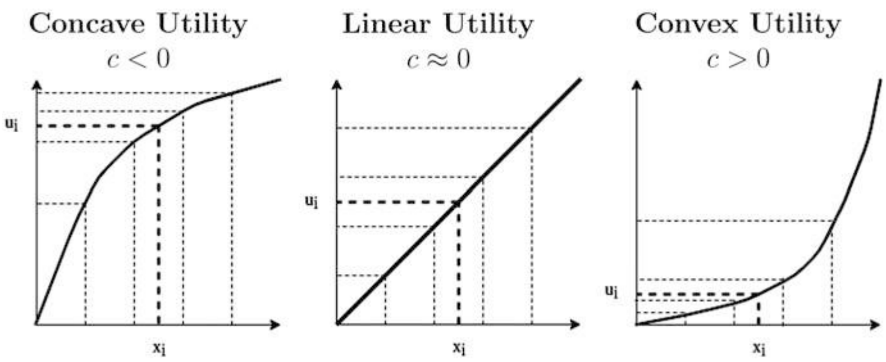
For minimization criteria:

$$x_k(a_i) = \frac{f_k(a_{max}) - f_k(a_i)}{f_k(a_{max}) - f_k(a_{min})} \tag{5.16}$$

Step 2: Integration of DM Attitude The attitude of the decision-maker is incorporated into the normalized decision matrix, as follows:

$$u_k(a_i) = \frac{1 - e^{cx_i}}{1 - e^c}$$

where c is an index that represents the attitude of the decision-maker.



Step 3: Utility Function The utility function is computed as follows:

$$U_i = \sum_{k=1}^{q} w_k u_k(a_i) \tag{5.17}$$

The MAUT final ranking is obtained by ordering the alternatives according to the decreasing values of the utility function.

Numerical Example Let us consider the example of Table 5.4, assuming that the DM's attitude is neutral and thus it does not affect the solution of the problem. After performing the normalization on the decision matrix as described in step 1, the normalized decision matrix is shown in Table 5.11.

$$x_1(a_1) = [f_1(a_1) - f_1(a_{min})]/[f_1(a_{max}) - f_1(a_{min})] = (20-10)/(20-10) = 10/10 = 1$$

$$x_3(a_3) = [f_3(a_3) - f_3(a_{min})]/[f_3(a_{max}) - f_3(a_{min})] = (10-5)/(35-5) = 5/30 \approx 0.16$$

$$x_1(a_4) = [f_1(a_4) - f_1(a_{min})]/[f_1(a_{max}) - f_1(a_{min})] = (10-10)/(20-10) = 0/10 = 0$$

Step 2 of the methodology is bypassed because we have assumed neutral DM's preference. Finally, the utility function as calculated by step 3 of the methodology is shown in Table 5.12.

$$U_1 = w_1 \times u_1(a_1) + w_2 \times u_2(a_1) + w_3 \times u_3(a_1) = 0.4 \times 1 + 0.1 \times 1 + 0.5 \times 0.5 = 0.75$$

$$U_2 = w_1 \times u_1(a_2) + w_2 \times u_2(a_2) + w_3 \times u_3(a_2) = 0.4 \times 0.5 + 0.1 \times 0.33 + 0.5 \times 0 \approx 0.23$$

Table 5.11 MAUT normalized decision matrix

Alternatives	Criterion 1	Criterion 2	Criterion 3
Sec1	1	1	0.5
Sec2	0.5	0.33	0
Sec3	0.2	0	0.16
Sec4	0	0.66	1

Table 5.12 MAUT Utility Function

	Utility Function
Sec1	0.75
Sec2	0.23
Sec3	0.16
Sec4	0.56

5.3.4 TOPSIS

TOPSIS (Hwang and Yoon, 1981) [82] is a multicriteria decision analysis method. The acronym TOPSIS stands for Technique for Order of Preference by Similarity to Ideal Solution. It is based on the geometric distance from the positive ideal solution (PIS) and the negative ideal solution (NIS). A good alternative has a short distance from the PIS and a long distance from the NIS.

The algorithm of the TOPSIS method is presented below:

Algorithm 5: TOPSIS Algorithm

---

input: n (alternatives), q (criteria), w_i (weights), $f_i(a_j)$ (evaluation table);
for *all alternatives* a_i, $i \in \{1, \ldots, n\}$ **do**
    for *all criteria* f_k, $k \in \{1, \ldots, q\}$ **do**
        compute normalized score $r_k(a_i)$
    end
end
for *all alternatives* a_i, $i \in \{1, \ldots, n\}$ **do**
    for *all criteria* f_k, $k \in \{1, \ldots, q\}$ **do**
        compute weighted normalized score $t_k(a_i)$
    end
end
compute positive ideal solution A^+
compute negative ideal solution A^-
for *all alternatives* a_i, $i \in \{1, \ldots, n\}$ **do**
    compute separation distance from positive ideal solution $S^+(i)$
    compute separation distance from negative ideal solution $S^-(i)$
    compute relative closeness to the positive ideal solution $C^-(i)$
end
$FinalRanking = \mathbf{SortDesc}(C)$
Result: FinalRanking

Step 1: Normalized Decision Matrix Given the $n \times q$ decision matrix, using the following normalization method, a new normalized decision matrix is calculated:

$$r_k(a_i) = \frac{x_k(a_i)}{\sqrt{\sum_{j=1}^{n} x_k(a_j)^2}} \tag{5.18}$$

Step 2: Weighted Normalized Decision Matrix In this step, the offsets are incorporated in order to quantify the relative importance of the different criteria. The weighted decision matrix is constructed by multiplying each element of each column of the normalized decision matrix by the offsets:

$$t_k(a_i) = r_k(a_i) \cdot w_k, \quad \text{where } w_k = W_k \Big/ \sum_{j=1}^{q} W_j,\ j = 1, 2, \ldots, q \tag{5.19}$$

Step 3: Positive and Negative Ideal Solution The positive ideal A^+ and the negative ideal A^- solutions are defined according to the weighted normalized decision matrix:

$$\begin{aligned} A^+ = \{\langle \min(t_k(a_i) \mid i = 1, 2, \ldots, n) \mid k \in J_-\rangle, \langle \max(t_k(a_i) \mid i = 1, 2, \ldots, n) \\ \mid k \in J_+\rangle\}\} \end{aligned} \tag{5.20}$$

$$\begin{aligned} A^- = \{\langle \max(t_k(a_i) \mid i = 1, 2, \ldots, n) \mid k \in J_-\rangle, \langle \min(t_k(a_i) \mid i = 1, 2, \ldots, n) \\ \mid k \in J_+\rangle\}\} \end{aligned} \tag{5.21}$$

where

$$J_+ = \{k = 1, 2, \ldots, q \mid k\}$$

for maximization criteria and

$$J_- = \{k = 1, 2, \ldots, q \mid k\}$$

for minimization criteria.

Step 4: Separation Distance from the Ideal and Non-Ideal Solution Let t_k^+ be the positive ideal value and t_k^- be the positive ideal value for criterion k. The separation distance ($L^2 - Distance$) of each alternative from the ideal and non-ideal solution is calculated:

$$S_i^+ = \sqrt{\sum_{k=1}^{q} (t_k(a_i) - t_k^+)^2}, \quad i = 1, 2, \ldots, n, \tag{5.22}$$

Table 5.13 TOPSIS normalized decision matrix

Alternatives	Criterion 1	Criterion 2	Criterion 3
Sec1	0.68	0.68	0.47
Sec2	0.51	0.41	0.12
Sec3	0.41	0.27	0.24
Sec4	0.34	0.54	0.84

Table 5.14 TOPSIS weighted decision matrix

Alternatives	Criterion 1	Criterion 2	Criterion 3
Sec1	0.27	0.07	0.24
Sec2	0.20	0.04	0.06
Sec3	0.16	0.03	0.12
Sec4	0.14	0.05	0.42

$$S_i^- = \sqrt{\sum_{k=1}^{q}(t_k(a_i) - t_k^-)^2}, \quad i = 1, 2, \ldots, n \tag{5.23}$$

Step 5: Relative Closeness to the Ideal Solution Finally, for each alternative the relative closeness to the ideal solution is computed. The TOPSIS final ranking is obtained by ordering the alternatives according to the decreasing values of the relative closeness scores.

$$C_i = S_i^- / (S_i^+ + S_i^-), \quad 0 \leq C_i \leq 1, \quad i = 1, 2, \ldots, n \tag{5.24}$$

Numerical Example Let us consider the example of Table 5.4. After performing the normalization on the decision matrix as described in step 1, the normalized decision matrix is shown in Table 5.13.

$$r_1(a_1) = x_1(a_1)/\sqrt{x_1(a_1)^2 + \cdots + x_1(a_4)^2} = 20/\sqrt{20^2 + 15^2 + 12^2 + 10^2} = 0.68$$

$$r_2(a_3) = x_2(a_3)/\sqrt{x_2(a_1)^2 + \cdots + x_2(a_4)^2} = 5/\sqrt{5^2 + 3^2 + 2^2 + 4^2} = 0.27$$

The following step involves the incorporation of the weights. Therefore, the weighted decision matrix is shown in Table 5.14.

$$t_1(a_1) = r_1(a_1) \times w_1 = 0.68 \times 0.4 = 0.27$$

$$t_2(a_3) = r_2(a_3) \times w_2 = 0.27 \times 0.1 \approx 0.03$$

The ideal (and the negative ideal) solution can be found if we select the best (and the worst) alternative for every criterion. Therefore:

$$A^+ = \{0.27, 0.07, 0.42\}$$

$$A^- = \{0.14, 0.03, 0.06\}$$

Table 5.15 TOPSIS separation distances and relative closeness to the ideal solution

Alternatives	S^+	S^-	Relative Closeness
Sec1	0.18	0.23	0.56
Sec2	0.37	0.07	0.16
Sec3	0.32	0.07	0.17
Sec4	0.14	0.36	0.73

Finally, in Table 5.15 we show the separation distance from the ideal and the negative ideal solution, as well as the relative closeness to the ideal solution.

$$S_1^+ = \sqrt{(0.27 - 0.27)^2 + (0.07 - 0.07)^2 + (0.24 - 0.42)^2} = 0.18$$

$$S_1^- = \sqrt{(0.27 - 0.14)^2 + (0.07 - 0.03)^2 + (0.24 - 0.06)^2} = 0.23$$

$$C_1 = S_1^- / (S_1^+ + S_1^-) = 0.23/(0.18 + 0.23) = 0.56$$

5.3.5 Cumulative Ranking

After the application of the four MCDA methods, four ranking lists of the alternatives have been formulated. However, the decision-maker should be provided with a final ranking in order to select the k-best securities among them. The suggested methodology to combine the four rankings is the weighted average measure. More specifically, each ranking method is provided with a weighting factor w_k, $k = \{1, 2, 3, 4\}$. The cumulative ranking index CR_i for alternative i is calculated as follows:

$$CR_i = \sum_{k=1}^{4} w_k r_k \tag{5.25}$$

where r_k represents the ranking of alternative i in method k.

5.4 Phase II: Multiobjective Portfolio Optimization

In the second phase let us introduce the concept of portfolio optimization. This section cures the problem of capital allocation to the selected securities. Portfolio optimization is the process of determining the best combination of the weighting factors of securities with the goal to minimize risk and maximize the profit.

The methodological framework of the first phase is depicted in Figure 5.4

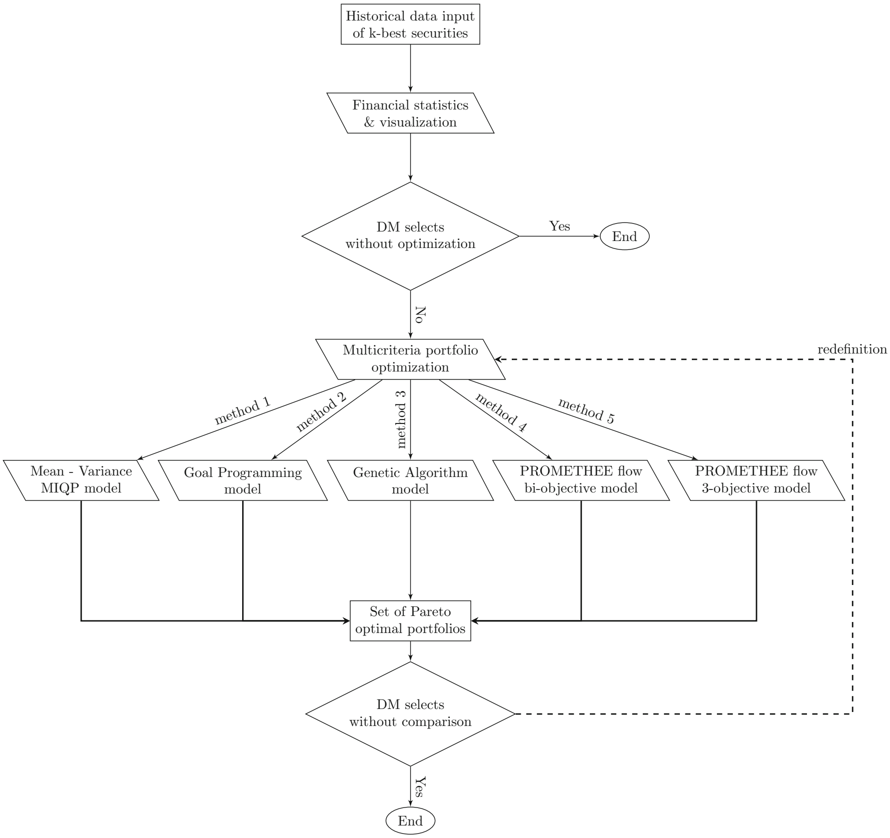


Fig. 5.4 Methodological framework for Phase II

The developed methodologies that will be discussed and compared in the following paragraphs are

1. implementation of the conventional mean–variance model, as well as a mixed-integer variation of the mean–variance method.
2. implementation of a goal programming optimization model.
3. implementation of a genetic algorithm based on historical data.
4. implementation of bi-objective optimization problem which involves the PROMETHEE net flow, as well as a MOIP variation of this problem.
5. implementation of a 3-objective optimization problem which involves the PROMETHEE net flow combined with two additional objective functions.

5.4.1 Mean–Variance MIQP Model

The conventional formulation of the portfolio optimization problem was initially expressed as a nonlinear bi-criteria optimization problem. According to Markowitz [112] the portfolio expected return should be maximized and the portfolio risk should be minimized. The risk is quantified as the variance of portfolio returns, resulting in a quadratic programming problem.

Let $E(R_i)$ be the expected return and w_i the weighting factor of security i. The first objective concerns the portfolio expected return and is expressed as follows:

$$\max_w E(R_p) = \sum_{i=1}^{m} w_i E(R_i)$$

where m is the total number of securities. Let σ_{ij} be the covariance between securities i and j. The second objective concerns the portfolio risk which is expressed as follows:

$$\min_w \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} w_i w_j \sigma_{ij}$$

Moving to the model's set of constraints, the corresponding expression for capital completeness is introduced:

$$\sum_{i=1}^{m} w_i = 1$$

while the restriction concerning no short sales allowance is

$$w_i \geq 0$$

The above equations constitute a bi-objective quadratic optimization problem which is presented below:

$$\begin{aligned}
&\max_w \; E(r_P) = \sum_{i=1}^{m} w_i E(r_i) \\
&\min_w \; \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} w_i w_j \sigma_{ij} \\
&\text{s.t.} \sum_{i=1}^{m} w_i = 1 \\
&\qquad w_i \geq 0, \qquad\qquad i = 1, 2, \ldots, m
\end{aligned} \tag{5.26}$$

The problem is solved parametrically for a predefined parameter of the portfolio expected return. Let R be the portfolio expected return. The problem is transformed into a linear programming problem with an additional restriction concerning the expected return, which is presented below:

$$\begin{aligned} \min_{w} \sigma_P^2 &= \sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} w_i w_j \sigma_{ij} \\ \text{s.t.} \sum_{i=1}^{m} w_i E(r_i) &= R \\ \sum_{i=1}^{m} w_i &= 1 \\ w_i &\geq 0, \qquad i = 1, 2, \ldots, m \end{aligned} \tag{5.27}$$

In this section, a variation of the conventional mean–variance model is developed (Xidonas & Mavrotas, 2012) [215]. The model is equipped with binary variables b_i, in order to control the existence of each security in the portfolio. More specifically, if $b_i = 1$, the i_{th} security participates in the portfolio, else if $b_i = 0$, it does not. The use of binary variables allows the direct determination of the number of securities in the portfolio, producing the following cardinality constraint equation:

$$S_L \leq \sum_{i=1}^{m} b_i \leq S_U$$

where S_L and S_U are the minimum and maximum number of securities allowed to participate in the portfolio.

Moreover, the diversification of the portfolio can be supported constraining the upper bound of each security weight. In order to determine the lower and upper weighting factor of each security the following restrictions are introduced:

$$w_i - W_L \times b_i \geq 0, \quad i = 1, 2, \ldots, m$$

$$w_i - W_U \times b_i \leq 0, \quad i = 1, 2, \ldots, m$$

where W_L and W_U are the minimum and maximum security weights that are allowed in the portfolio.

Thus, the following multiobjective integer programming (MOIP) problem is formulated:

$$
\begin{aligned}
&\max_{w}\ E(r_P) = \sum_{i=1}^{m} w_i E(r_i) \\
&\min_{w}\ \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} w_i w_j \sigma_{ij} \\
&\text{s.t.} \qquad \sum_{i=1}^{m} b_i \le S_U \\
&\qquad\qquad \sum_{i=1}^{m} b_i \ge S_L \\
&\qquad\qquad \sum_{i=1}^{m} w_i = 1 \\
&\qquad w_i - W_L \times b_i \ge 0 \qquad i = 1, 2, \ldots, m \\
&\qquad w_i - W_U \times b_i \le 0 \qquad i = 1, 2, \ldots, m
\end{aligned}
\tag{5.28}
$$

Similarly, the solution is determined parametrically for a predefined parameter of the portfolio expected return. The problem is transformed into a mixed-integer quadratic programming (MIQP) problem with an additional restriction for the expected return, which is presented below:

$$
\begin{aligned}
&\min_{w}\ \sigma_P^2 = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} w_i w_j \sigma_{ij} \\
&\text{s.t.} \quad \sum_{i=1}^{m} w_i E(r_i) = R \\
&\qquad\qquad \sum_{i=1}^{m} b_i \le S_U \\
&\qquad\qquad \sum_{i=1}^{m} b_i \ge S_L \\
&\qquad\qquad \sum_{i=1}^{m} w_i = 1 \\
&\qquad w_i - W_L \times b_i \ge 0 \qquad i = 1, 2, \ldots, m \\
&\qquad w_i - W_U \times b_i \le 0 \qquad i = 1, 2, \ldots, m
\end{aligned}
\tag{5.29}
$$

The formulated problem can be solved parametrically considering the parameter R, thus producing the efficient frontier of solutions.

5.4.2 Goal Programming Model

Another approach to the portfolio optimization problem is the development of a goal programming model. As discussed in the previous chapter, goal programming is a multiobjective optimization method which is used to handle problem of conflicting objective functions.

The decision variables of the goal programming problem will be the weighting factor w of each security. Let w_i be the weighting factor of the i_{th} security. The following goals are defined:

1. The beta index of the portfolio β_P, which is defined as the weighted sum of the individual beta index of each security, is given the target value β_G.

$$\beta_P = \sum_{i=1}^{m} w_i \times \beta_i$$

2. The portfolio dividend yield, which is defined as the weighted sum of the individual dividend yield of each security, is given the target value DY_G.

$$DY_P = \sum_{i=1}^{m} w_i \times DY_i$$

3. The portfolio PROMETHEE flow, which is defined as the weighted sum of the individual flow of each security, is given the target value ϕ_G

$$\phi_P = \sum_{i=1}^{m} w_i \times \phi_i$$

The model is equipped with binary variables b, in order to control the existence of each security in the portfolio, producing the following cardinality constraint equation:

$$S_L \leq \sum_{i=1}^{m} b_i \leq S_U$$

where S_L and S_U are the minimum and maximum number of securities allowed to participate in the portfolio.

In order to determine the lower and upper weighting factor of each security the following restrictions are introduced:

$$w_i - W_L \times b_i \geq 0, \quad i = 1, 2, \ldots, m$$

$$w_i - W_U \times b_i \leq 0, \quad i = 1, 2, \ldots, m$$

where W_L and W_U are the minimum and maximum security weights that are allowed in the portfolio.

Introducing the deviational (or slack) variables d_i^+, d_i^- the problem is formulated as follows:

$$
\begin{aligned}
\min_{d^+,d^-} \quad & \frac{w_1^+ d_1^+ + w_1^- d_1^-}{\beta_G} + \frac{w_2^+ d_2^+ + w_2^- d_2^-}{DY_G} + \frac{w_3^+ d_3^+ + w_3^- d_3^-}{\phi_G} \\
\text{s.t.} \quad & \sum_{i=1}^{m} w_i \beta_i + d_1^- - d_1^+ = \beta_G \\
& \sum_{i=1}^{m} w_i DY_i + d_2^- - d_2^+ = DY_G \\
& \sum_{i=1}^{m} w_i \phi_i + d_3^- - d_3^+ = \phi_G \\
& \sum_{i=1}^{m} b_i \leq S_U \\
& \sum_{i=1}^{m} b_i \geq S_L \\
& \sum_{i=1}^{m} w_i = 1 \\
& w_i - W_L \times b_i \geq 0 \qquad i = 1, 2, \ldots, m \\
& w_i - W_U \times b_i \leq 0 \qquad i = 1, 2, \ldots, m
\end{aligned}
\tag{5.30}
$$

where w^+, w^- are the weights of the deviational variables. Attention is needed not to confuse the weighting factor w of each security with the overachievement and underachievement weights w^+,w^- of the deviational variables.

5.4.3 Genetic Algorithm Model

In this paragraph we introduce an implementation of an alternative model applying a genetic algorithm. The philosophy of this problem differs from all the other, as the weighting factors are determined with the assistance of a market index. Additionally, another significant difference is that in this case there is a unique solution of the optimization problem, while the other problems result in a set of Pareto efficient solutions.

More specifically, let m be the market index and r_{im} the return of index m in period i. Let us define, also, m securities and r_{ij} the return of security j in period i. The portfolio return in period i is equal to:

$$r_{ip} = \sum_{j=1}^{m} w_j r_{ij} \tag{5.31}$$

where w_j is the proportion of the security j in portfolio p.

We say that *the constructed portfolio beats the market index* in period i if the following inequation applies:

$$r_{ip} \geq r_{im} \tag{5.32}$$

Therefore, the genetic algorithm takes as input the historical data for T periods and attempts to maximize the percentage of cases that the constructed portfolio beats the market index. This claim is quantified as follows:

$$\begin{aligned} \max_{w_i} \sum_{i=1}^{T} b_i \Big/ T & \\ \text{s.t.} \sum_{i=1}^{N} w_i & = 1 \\ w_i & \geq 0 \quad \forall i = 1, 2, \ldots N \end{aligned} \tag{5.33}$$

where b_i is a binary variable that takes the value 1 if $r_{ip} \geq r_{im}$ in period i, else it takes the value 0.

The selected genetic algorithm that we use in the proposed methodology is called *differential evolution*. Differential evolution [193] is a stochastic population based method developed by Storn and Price, which is used in order to find the global minimum of a multivariate function. Every time that the algorithm examines the population, it mutates each candidate solution by mixing it with other candidate solutions in order to create a trial candidate. The most common strategy to create a trial candidate is the *best1bin* strategy, where two members of the population are randomly chosen and their difference is used to mutate the best member b_0:

$$b' = b_0 + mutation * (population[i] - population[j]) \tag{5.34}$$

Then, a trial vector is constructed which is filled with parameters either from b' or the original candidate, generated with a binomial distribution (meaning that we generate a random number between 0 and 1; if the number is less than a predefined constant, then the parameter is filled from b', otherwise it is filled from the original candidate. After the construction of the trial candidate, its fitness is assessed. If the trial candidate is better than the original candidate, it replaces the original candidate. If it is also better than the best overall candidate, it also replaces that. By default the best solution vector is updated continuously within a single iteration. Finally, the possibility of finding a global minimum is improved by increasing the population size, which results in widening the search radius, but slowing the convergence of the algorithm.

Conclusively, the genetic algorithm provides the optimal portfolio proportions, such that the percentage that the constructed portfolio results in better return than the market index is maximized.

5.4.4 MOIP PROMETHEE Flow Model

PROMETHEE Flow Bi-Objective Model

In this paragraph, an alternative approach to the classic mean–variance model is presented. This approach connects the concept of the PROMETHEE method of multicriteria decision analysis with a measure of risk, in this case beta index. The beta index of a portfolio is the average of the beta indexes of the participating securities. Let ϕ_i represent the PROMETHEE net flow of the i_{th} security. The first objective function of the problem involves the maximization of the average net flow:

$$max_w \quad \phi_P = \sum_{i=1}^{m} w_i \phi_i$$

The second objective of the problem involves the minimization of portfolio beta index:

$$min_w \quad \beta_P = \sum_{i=1}^{m} w_i \beta_i$$

Therefore, adding to the model the constraints of capital completeness and no short sales allowance, a bi-objective linear programming optimization problem is formulated as follows:

$$\begin{aligned} \max_w \ \phi_P &= \sum_{i=1}^{m} w_i \phi_i \\ \min_w \ \beta_P &= \sum_{i=1}^{m} w_i \beta_i \\ \text{s.t.} \sum_{i=1}^{m} w_i &= 1 \\ w_i \geq 0, &\quad i = 1, 2, \ldots, m \end{aligned} \tag{5.35}$$

If we introduce binary variables to the above problem, accordingly to the extension of mean–variance model, a bi-objective integer programming problem is formulated as follows:

$$
\begin{aligned}
\max_{w} \ \phi_P &= \sum_{i=1}^{m} w_i \phi_i \\
\min_{w} \ \beta_P &= \sum_{i=1}^{m} w_i \beta_i \\
\text{s.t.} \qquad & \sum_{i=1}^{m} b_i \le S_U \\
& \sum_{i=1}^{m} b_i \ge S_L \\
& \sum_{i=1}^{m} w_i = 1 \\
w_i - W_L \times b_i &\ge 0 \quad i = 1, 2, \ldots, m \\
w_i - W_U \times b_i &\le 0 \quad i = 1, 2, \ldots, m
\end{aligned}
\tag{5.36}
$$

The above problems can be solved parametrically, transforming one objective function into an additional constraint, accordingly to the conventional mean–variance model.

PROMETHEE Flow 3-Objective Model

Finally, in this paragraph a 3-objective variation of the above model is introduced, in order to incorporate another objective function in the portfolio optimization problem.

Without loss of generality, let PROMETHEE flow be the first objective function. Alternatively, any other method could be introduced, such as ELECTRE III, MAUT, or TOPSIS. We selected the PROMETHEE net flow, because of PROMETHEE's capability to use a variety of functions depending on the criterion. Therefore, the first goal is to optimize the net flow of PROMETHEE as it was introduced in Phase A.

We propose the second objective to be the beta index in order to incorporate a measure of risk to the model. Finally, let us introduce the dividend yield as a third objective function.

Additionally, the problem could be equipped with integer decision variables in order to control the number of securities with non-zero proportion to the portfolio. Based on the above observations the optimization problem is formulated as follows:

$$\begin{aligned}
\max_{w}\ \phi_P &= \sum_{i=1}^{m} w_i \phi_i \\
\min_{w}\ \beta_P &= \sum_{i=1}^{m} w_i \beta_i \\
\max_{w}\ DY_P &= \sum_{i=1}^{m} w_i DY_i \\
\text{s.t.} \qquad & \sum_{i=1}^{m} b_i \le S_U \\
& \sum_{i=1}^{m} b_i \ge S_L \\
& \sum_{i=1}^{m} w_i = 1 \\
w_i - W_L \times b_i &\ge 0 \quad i = 1, 2, \ldots, m \\
w_i - W_U \times b_i &\le 0 \quad i = 1, 2, \ldots, m
\end{aligned} \tag{5.37}$$

This is a multiobjective programming problem in 3 dimensions with integer variables. It is obvious that the computational complexity of the above problem becomes huge, especially if the number of securities is significantly large. A variety of methodologies have been proposed for problems like this such as the e-constraint method, which faces the problem as a 1-objective optimization problem, transforming the remaining objectives into constraints. However, in this paragraph a methodology based on goal programming and the MINIMAX objective is proposed to solve this MOLP problem.

The first step of the methodology is to solve the model to find the solution that minimizes each objective function ignoring the other objective function. If we solve the problem for all objective functions, we obtain the optimal value for each objective, respectively.

The next step is to formulate the goal programming problem. The target value for each objective function is set equal to the optimal value calculated in the previous step. The percentage deviation from this target can be computed as follows:

$$t = \frac{actual\ value - target\ value}{target\ value} \tag{5.38}$$

for goals derived from minimization objectives,

$$t = \frac{target\ value - actual\ value}{target\ value} \tag{5.39}$$

for goals derived from maximization objectives.

Therefore, having determined the goals of the GP model, the last step involves the configuration of the objective function. The implementation of the objective function is made with the introduction of a MINIMAX variable Q which should be minimized. If w_i is the offset for the i_{th} objective function, the goal is to minimize the maximum of $w_i t_i$. The above claim is expressed with the following mathematical equation:

$$\begin{aligned} \min\ & Q \\ \text{s.t. } & w_1 t_1 \leq Q \\ & w_2 t_2 \leq Q \\ & w_3 t_3 \leq Q \end{aligned} \tag{5.40}$$

Thus, a set of Pareto optimal solutions derives from the adjustment of the weighting factors w_i.

5.5 Conclusions

In this chapter there was a complete presentation of the proposed methodology, which aspires to incorporate all the parameters of the problem of efficient portfolio management. The proposed methodology was split into two phases: in the first phase we introduced an integrated methodology for the problem of security selection and in the second phase we presented a series of methodologies which address the problem of portfolio optimization.

For completeness reasons, it is necessary to make a quick discussion about the final decision problematic, concerning the selection of the most suitable portfolio from the set of efficient portfolios. Given a Pareto efficient set of candidate portfolios, the problem lies to the determination of the most appropriate portfolio. It is obvious that the most significant parameter of the problem is the profile of the decision-maker. The decision-maker's profile creates a clear picture about how a person makes a decision and determines the way that he should be supported in decision-making. For instance, a risk-averse decision-maker would probably exclude all the portfolios that result in increased portfolio risk, while on the contrary an aggressive decision-maker would select one of the riskiest portfolios of the efficient set. Subsequently, the decision-maker might determine the suitable portfolio without any additional support.

However, in case that the investor has not reached to a final decision a methodological framework for decision support of the final phase is developed. The problem of selecting the most dominant portfolio of all the feasible ones can be solved as a discrete MCDA problem, where the alternatives are all the efficient portfolios and the criteria can be set in communication with the decision-maker. Therefore, the methodological framework that was used in Phase I for security selection can be also used to this problem for the determination of the most suitable portfolio. After the application of the MCDA methods a final ranking of the portfolios is obtained and finally, the investor can select the most appropriate one.

Chapter 6
Information System in Python

6.1 Introduction

The presentation of the proposed methodology emphasized the need for modern information systems in order to implement the necessary methods. The purpose of the information system is to implement efficiently the algorithms described in the previous section in order to support the decision-making process.

The information system consists of four subsystems. The first subsystem includes an implementation of the multicriteria decision support methods that are used in the proposed methodology. The second subsystem supports financial statistics calculation. The third subsystem implements the multiobjective programming methods for portfolio optimization. Finally, the fourth subsystem assists the evaluation process, offering a visualization of the efficient portfolios.

In addition, the first subsystem is deployed as a web application. The specific platform offers a friendly graphical user interface *GUI* and offers efficient implementations of specific MCDA methods, providing extensive solutions for a wide range of multicriteria problems.

The chapter is organized as follows: In Section 6.2, we present the architecture of the system, the programming language, and the libraries which were utilized. In Section 6.3, we introduce the interaction diagrams aiming to familiarize the reader with the flowchart of the information system. In Section 6.4, we present the MCDA platform, the input that is needed and the output data that it produces. Finally, in Section 6.5 there is a detailed presentation of the Python source code, with specific examples and comments.

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_6

6.2 System Architecture and Attributes

In this section, there is a brief introduction to the programming tools which were used, as well as the libraries that supported the visualization of data and the optimization methods. The information system is developed in Python 3 programming language, which makes it available for Windows, Linux, and macOS operating systems.

6.2.1 Python 3.0 Programming Language

Python is a general-purpose, high level scripting programming language, which is widely used nowadays. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. Its main goal is to provide code readability and simplify complex concepts. It is an interpreted language, i.e. the steps of code compilation and execution are unified and the program can be directly executed from the source code. Additionally, it is platform independent as it can be used on Linux, Windows, Macintosh, and multiple other operating systems. Python can be used for a variety of tasks in many sectors including: (a) mathematics and physics, (b) quantitative finance and financial econometrics, (c) machine learning, neural networks and artificial intelligence, (d) big data applications and data engineering, (e) network security, prototyping applications, and (f) enterprise and business applications, web frameworks and applications, etc.

More specifically, Python 3.0 (also called "Python 3000" or "Py3K") was released on December 3, 2008. It has a wide variety of advantages, the most important of which are the following: (a) it incorporates an extensive support of additional libraries, such as NumPy for numerical calculation and Pandas for data analysis, (b) it provides object-oriented utilities, and is portable and interactive, (c) it is an open source language, with a developed community, and (d) it is considered an easy to learn programming language, providing user-friendly data structures.

In the following paragraphs we make a quick introduction to the libraries that were used for the implementation of the model. The main libraries which support the development of the information system are matplotlib, pandas, NumPy, and MIP.

Matplotlib [79] is a Python plotting library which produces high quality figures in a variety of formats, such as plots, histograms, power spectra, bar charts, errorcharts, scatter-plots, etc. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, and web application servers. It was originally written by John D. Hunter, providing an interface with close resemblance to MatLab and it has an active development community.

Pandas [122] is an open source library which provides high performance, useful data structures, and data analysis tools for the Python programming language. Some of the most important included features are an efficient DataFrame object for data manipulation with integrated indexing, tools for reading and writing data between different formats (CSV, text files, Microsoft Excel, etc.), and flexible reshaping functionality of data sets. It incorporates significant optimizations providing high performance and it has a wide variety of uses in academic and commercial fields.

NumPy [134] is the fundamental library for scientific computing with Python, as it contains various useful tools. The most important functionalities are the N-dimensional array object, the tools for importing C/C++ and Fortran code, as well a variety of function, such as linear algebra, Fourier transform, and random number capabilities. Additionally, NumPy can also be used as an efficient multidimensional container of generic data, allowing NumPy to integrate with a wide variety of databases with very high speed.

MIP is a library of Python tools for the modeling and solution of mixed-integer linear programming problems. Some of the main functionalities provided by MIP are the following: Firstly, high level modeling capability offering the opportunity of easy implementation of linear relations, as in high level languages. Besides, the Python MIP package is deeply integrated with many solvers, such as Branch-and-Cut and the commercial solver Gurobi (Fig. 6.1).

6.3 Interaction Diagrams

In this section, we present the basic UML interaction diagrams of the information system, in order to demonstrate how it can be used. Through these diagrams, the interaction between the decision-maker and the information system is reflected.

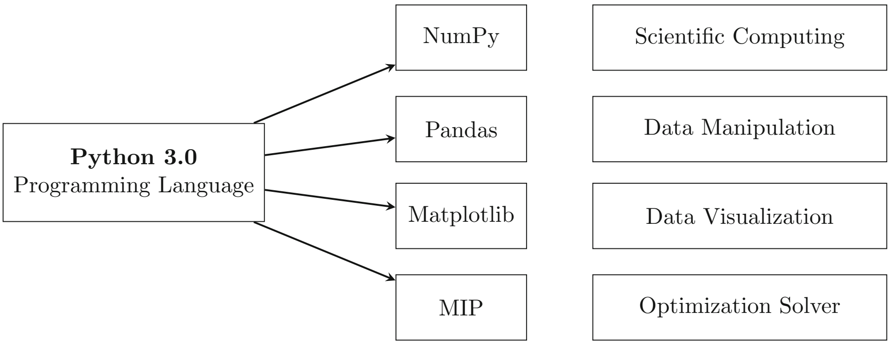


Fig. 6.1 Programming language and additional packages

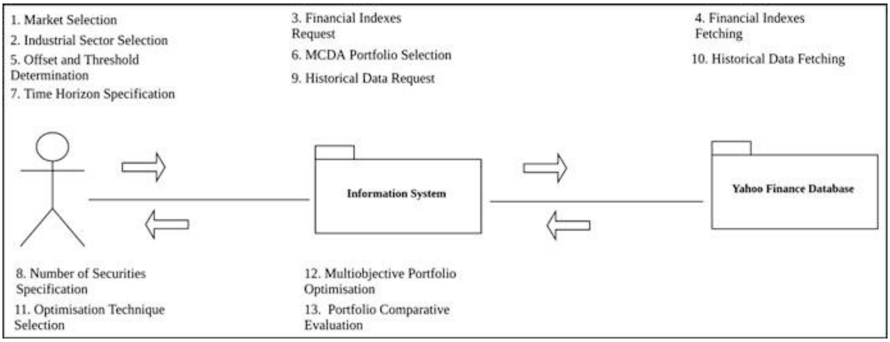


Fig. 6.2 UML Communication Diagram

6.3.1 Communication Diagram

A communication diagram describes the interactions between the various parts of the system in the form of sequenced messages. These diagrams are composed of a combination of sequence and use case diagrams incorporating both the static and the dynamic behavior of the information system. The communication diagram of the system is presented in Figure 6.2, where you can see the sequence of the events and the actions made from the different parts of the system in increasing order.

6.3.2 Use Case Diagram

A use case diagram is a graphic demonstration of the interactions among the elements of an information system. In the following diagram the three parts of the system—the decision-maker, the information system, and the Yahoo Finance Database—are described. As shown in Figure 6.3, the information system interacts with both the user and the Yahoo Finance Database. The user is able to perform the following actions: (a) Selection of the market, (b) determination of the offsets and the thresholds for the MCDA methods, (c) selection of the appropriate optimization methodology, (d) specification of the time horizon for the analysis, and (e) determination of the policy constraints for the optimization of the portfolio. The Yahoo Finance Database is involved in the process with the following attributes: (a) Secure connection with the information system, (b) integration of the financial indexes for the first phase of the methodology, and (c) incorporation of the historical data for the second phase of the methodological framework.

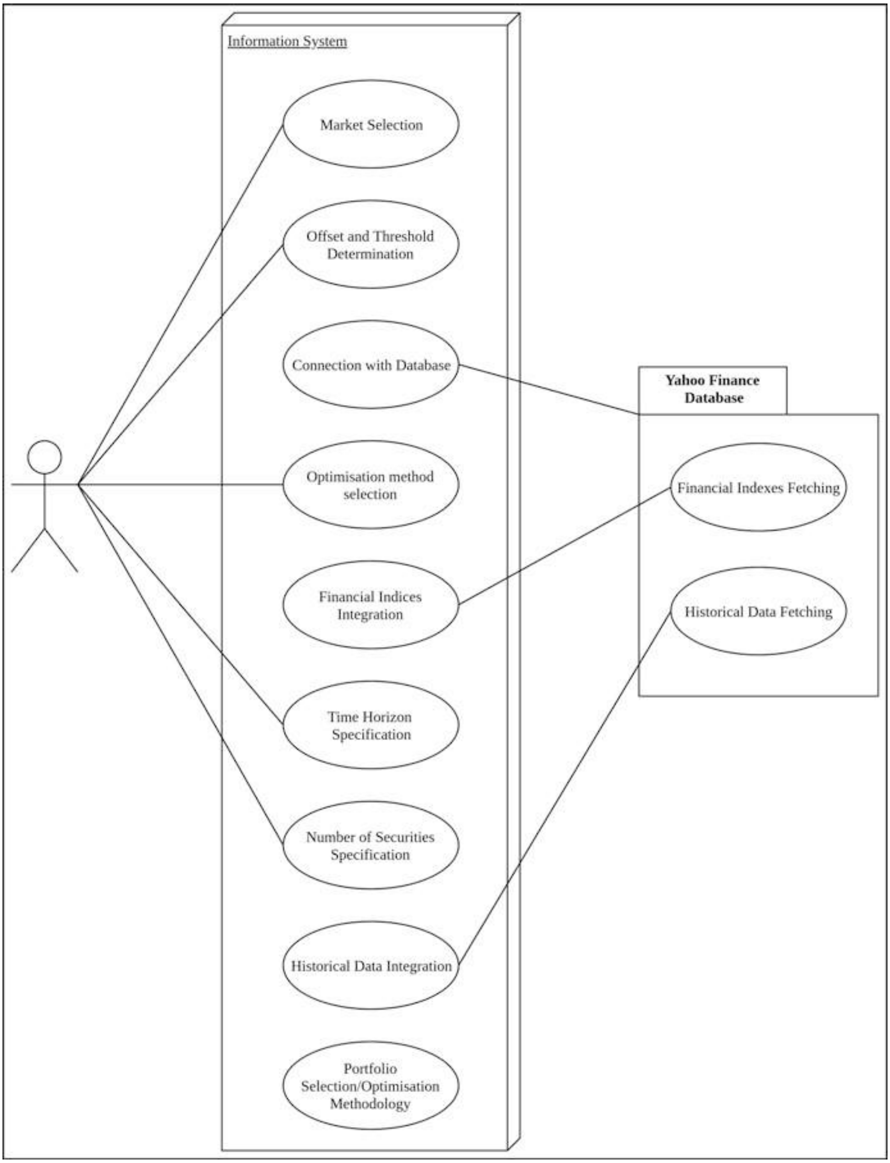


Fig. 6.3 UML Use Case Diagram

6.3.3 Sequential Diagram

A sequential diagram describes the interactions among each part of the system arranged in time sequence. It shows the sequence of messages which are exchanged between the different parts of the system (decision-maker, database, etc.), in order to achieve the functionality of each scenario. As presented in Figure 6.4, the procedure begins with the selection of the market and the industrial sector which are given as input from the user to the platform. The system communicates with the Yahoo Finance Database asking for the financial indexes which are used for the ranking methods. In case everything goes well, the system informs the user with a confirmation message and demands the values of the offsets and the thresholds. Therefore, when all the input data are gathered the optimal portfolio is calculated. Passing to the second phase of the process, the user determines the time horizon of the evaluation and the additional restrictions, such as number of securities, maximum percentage of every security, etc. The systems perform another communication with the database in order to fetch the historical data. Finally, the optimized portfolio is calculated according to the specified method from the user.

6.4 MCDA Platform

The first phase of the information system was deployed as a web application. This application includes an efficient implementation of a variety of multicriteria decision analysis methods, such as ELECTRE, PROMETHEE, MAUT, and TOPSIS. This project is implemented in one of the most used Python web frameworks called *Django*.

Django is a high-level Python free and open-source web framework which supports rapid development combined with modern design. It follows the conventional model-template-view web architecture and it is used for the creation of database-driven websites. Django framework is generally based on an MVC architecture, as it consists of an object-relational mapper (ORM) that mediates between data models. Additionally, it includes a relational database called *Model*, a subsystem for processing HTTP requests known as *View*, and a URL dispatcher called *Controller*.

In this paragraph, we introduce some of the most important features of the MCDA platform. Initially, the homepage of the system is designed in order to present the application's content. As you can see in Figure 6.5, the homepage offers a horizontal bar which contains all the MCDA methodologies that are implemented in the system. Therefore, the user can have an easy access to the desired method.

The user is able to select one of the supported MCDA methods and have access to the detailed information and the input prerequisites of the selected method. As seen in Figure 6.6, where we present the screen of the PROMETHEE method as an example, on the top of the screen we present some information about the method, which is followed by the input parameters. The input parameters are different for every method, therefore it is important to inform the user about which of them are necessary to be given as input. Finally, on the bottom of the page there is a field where the user inserts the .csv file of his choice.

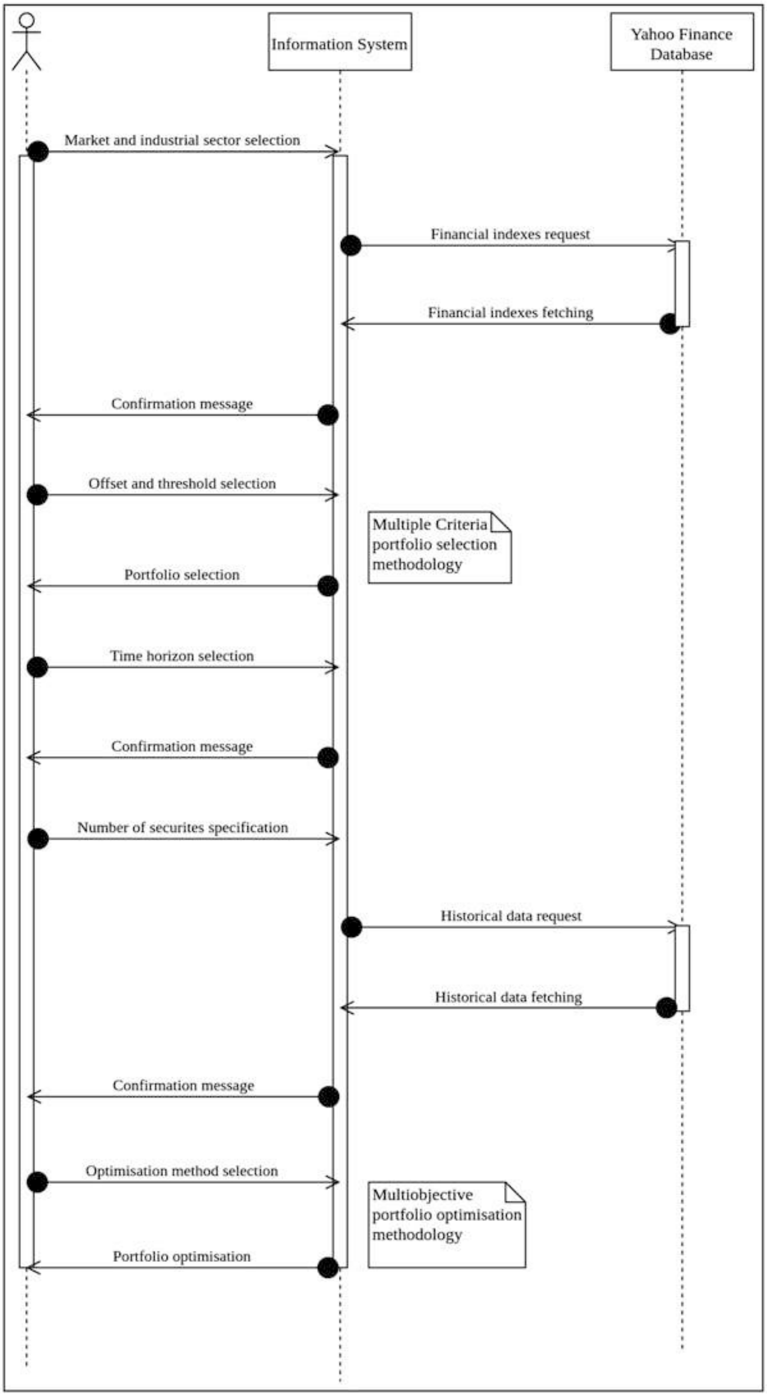


Fig. 6.4 UML Sequential Diagram

Fig. 6.5 Homepage screen

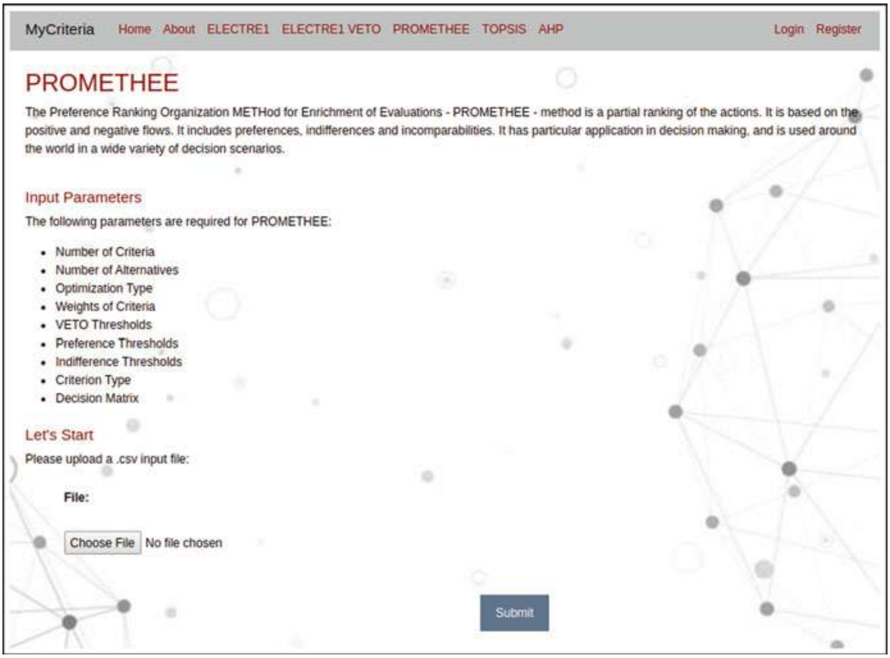


Fig. 6.6 Individual method page screen

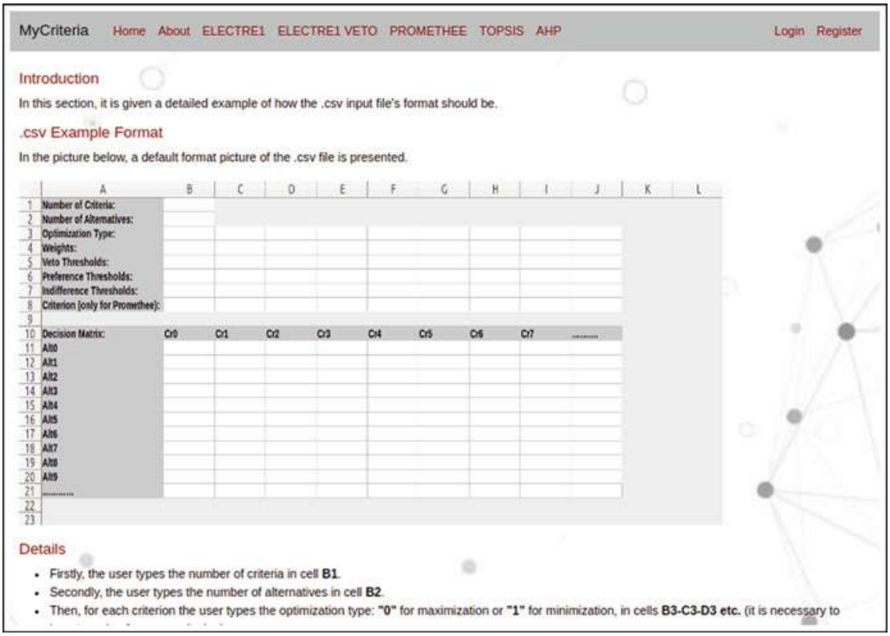


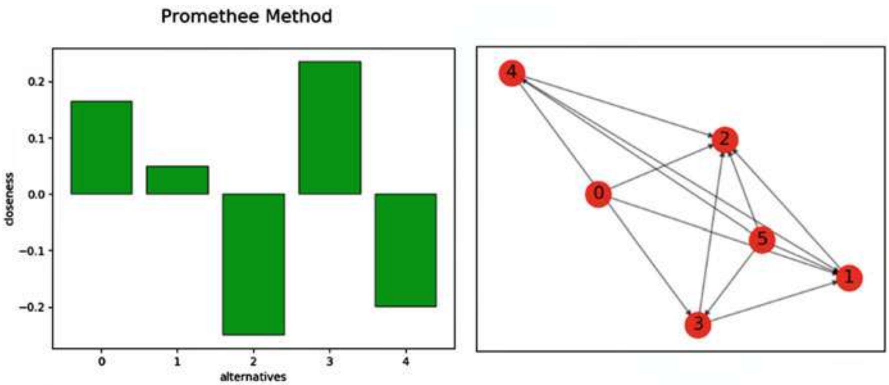
Fig. 6.7 About page screen

The platform supports all necessary methods for the proposed methodology, which are (a) ELECTRE III, (b) PROMETHEE II, (c) MAUT, and (d) TOPSIS. Except from these ranking methods, the platform also supports two choice methods: ELECTRE I and ELECTRE VETO, one classification method: ELECTRE TRI, as well as an implementation of weight calculation with entropy. Additionally, there is an *About* page which refers to the format of the necessary input. In Figure 6.7 we present the About page where there is a sample of the .csv input file. In the first row the user should insert the number of the alternatives and the number of the criteria. In the following lines, he should determine other additional information such as the offset of each criterion and whether it is a maximization or minimization criterion. According to the method he selects, he might also need to provide some thresholds such as preference, indifference, or veto thresholds. More specifically, all methods require to import the weights, the number of alternative and criteria, and the optimization type (maximization or minimization). Furthermore, PROMETHEE requires defining which criterion to use, as well as preference and indifference thresholds. ELECTRE 3 and ELECTRE TRI also require preference, indifference, and veto thresholds. Finally, the user should provide the evaluation matrix, which incorporates the value of every alternative in each criterion.

The input can be given in *.csv* or *.xls* format to the platform. Except from the general information (alternatives, criteria, decision matrix, and weights), different additional information should be imported according to the method. For example,

(a) ELECTRE I (b) TOPSIS (i) (c) TOPSIS (ii)

Fig. 6.8 Results Presentation Screen



(a) Ranking method (b) Choice method

Fig. 6.9 Results Visualization Screen

in PROMETHEE methods, it is necessary the criterion type and the appropriate thresholds to be determined. After the procedure of committing the input file to the platform, the next stage incorporates the presentation of the results. In Figure 6.8 we see the output screen for the application of some of the supported methods.

Finally, a visualization of the results can be produced by clicking the corresponding button at the results page. The visualization depends on the method and it is adjusted to the method category. For instance, for a ranking method a bar-plot with the final ranking of the alternatives is produced, while for a choice method the dominance graph is alternatively produced, showing which alternatives dominate over the others as shown in Figure 6.9. More specifically, the four ranking that are also used in the proposed methodology produce as output the final ranking of the alternatives (accompanied with the intermediate steps of the methodology). On the other side, the ELECTRE I and ELECTRE I Veto methods, which are choice methods, produce as output the set of the kernel with the most dominant solutions. Finally, the output of ELECTRE TRI, which is a classification method, is the class per alternative.

6.5 Source Code Presentation

In this section we present the most important parts of the source code of the information system. The presentation is divided into two parts: (a) multicriteria portfolio selection and (b) multiobjective portfolio optimization. In the beginning of the first part, we show how to import the necessary Python libraries and how to define the input of the MCDA methods. The first part includes the implementation of the four MCDA ranking methods (MAUT, TOPSIS, ELECTRE III, and PROMETHEE II). The Python implementation of these methods is organized in small sections of code. Therefore, the methods are presented step by step, in accordance to the theoretical presentation of the methods in Chapter 5. In the second part, we present the implementation of the multiobjective portfolio optimization techniques. In the beginning of this part, we make a quick introduction on how to connect with the Yahoo Finance API and how to import the historical data for the selected securities. Subsequently, we describe some fundamental financial calculations with Python, such as stock return and statistical indexes of the securities. Finally, we attempt a quick presentation of the Python solvers that were used for portfolio optimization.

A complete set of Python scripts including all the methods that we have used in the book along with a series of educational step by step examples can be found in the following url: https://github.com/epu-ntua/Multicriteria-Portfolio-Construction-with-Python.

Phase I: Multicriteria Portfolio Selection

Initially, we import all necessary libraries, including NumPy, matplotlib, and pandas which were presented in the previous section.

[1]:
```
import csv
import numpy as np
import matplotlib.pyplot
import math
import pandas as pd
from matplotlib import rc
import matplotlib.pylab as plt
```

```
rc('font', **{'family': 'serif', 'serif': ['Computer␣
↪Modern']})
rc('text', usetex=True)
```

After importing the necessary libraries, we should define the input information such as the criteria, the alternatives, the decision matrix, etc. The optimization type is set to 0 if it is an maximization criterion and 1 if it is a minimization criterion. Additionally, we provide the preference, indifference, and veto thresholds, as well as the desired criterion for the PROMETHEE method.

[2]:
```
criteria = 8
alternatives = 6
weights = [0.125, 0.125, 0.125, 0.125, 0.125, 0.125,␣
↪0.125, 0.125]
optimizationType = [1, 0, 0, 1, 0, 0, 0, 0]
criterion = [5, 5, 5, 5, 5, 1, 5, 5]
vetoThreshold = [15, 15, 50, 0.8, 3, 3, 30, 30]
preferenceThreshold = [10, 10, 20, 0.4, 2, 2, 15, 15]
indifferenceThreshold = [4, 4, 8, 0.2, 1, 0, 5, 5]
criterionName = ['P/E Ratio', 'EPS', 'Revenue (B)',␣
↪'Beta', 'Dividend Yield', 'Monthly', 'YTD (%)', '1␣
↪Year']
companyName = ['Accenture', 'Northrop Grumman',␣
↪'IBM', 'Motorola', 'MSCI', 'Oracle']
decisionMatrix = [[25.05, 7.36, 43.22, 1.05, 1.74, 5,␣
↪30.37, 15.3],
                  [18.15, 20.26, 32.9, 0.8, 1.44, 5,␣
↪50.57, 21.95],
                  [11.75, 12.01, 77.86, 1.36, 4.59,␣
↪4, 24.16, 0.2],
                  [31.39, 5.45, 7.63, 0.59, 1.33, 5,␣
↪47.78,41.39],
                  [33.64, 6.62, 1.48, 1.12, 1.22, 5,␣
↪51.02, 44.54],
                  [18.17, 3, 39.53, 1.15, 1.76, 5, 22.
↪9, 17.12]
                 ]
```

After we transform all the data from the .csv file to a dataframe, we present the source code which implements the 4 MCDA ranking methods for portfolio selection step by step.

[3]:
```
df = pd.DataFrame.from_records(decisionMatrix,␣
↪index=companyName ,columns=criterionName)
```

6.5.1 *MAUT*

Step 1: Initially, we calculate the normalized decision matrix. For the normalization we calculate the maximum and minimum value per criterion in the variables `maxValue`, `minValue`. Afterwards, we display the produced matrix with the `display` command.

[4]:
```
maxValue = np.max(decisionMatrix, axis = 0)
minValue = np.min(decisionMatrix, axis = 0)

normalizedMatrix = [[0 for y in range(criteria)] for x in range(alternatives)]
for i in range(alternatives):
        for j in range(criteria):
                if optimizationType[j] == 0:
                        normalizedMatrix[i][j] = (decisionMatrix[i][j] - minValue[j])*1.0 / (maxValue[j] - minValue[j])
                elif optimizationType[j] == 1:
                        normalizedMatrix[i][j] = (maxValue[j] - decisionMatrix[i][j])*1.0 / (maxValue[j] - minValue[j])

df = pd.DataFrame.from_records(normalizedMatrix, index=companyName ,columns=criterionName)
display(df)
```

=============== Normalized decision Matrix ===============

Company	P/E	EPS	Rev(B)	Beta	D.Yield	Techn	YTD(%)	1 Year
Accenture	0.392	0.252	0.546	0.402	0.154	1.0	0.265	0.340
Northrop	0.707	1.000	0.411	0.727	0.065	1.0	0.983	0.490
IBM	1.000	0.522	1.000	0.000	1.000	0.0	0.044	0.000
Motorola	0.102	0.141	0.080	1.000	0.032	1.0	0.884	0.928
MSCI	0.000	0.209	0.000	0.311	0.000	1.0	1.000	1.000
Oracle	0.706	0.000	0.498	0.272	0.160	1.0	0.000	0.381

Step 2: As you can see in the above results, the values of the initial decision matrix have been normalized to 0-1. After the normalization process the utility score is calculated by multiplying the normalized matrix with the weight of every criterion. Therefore, we calculate the `utilityScore` array as follows:

[5]:
```
utilityScore = [0 for x in range(alternatives)]

for i in range(alternatives):
        tempSum = 0
        for j in range(criteria):
```

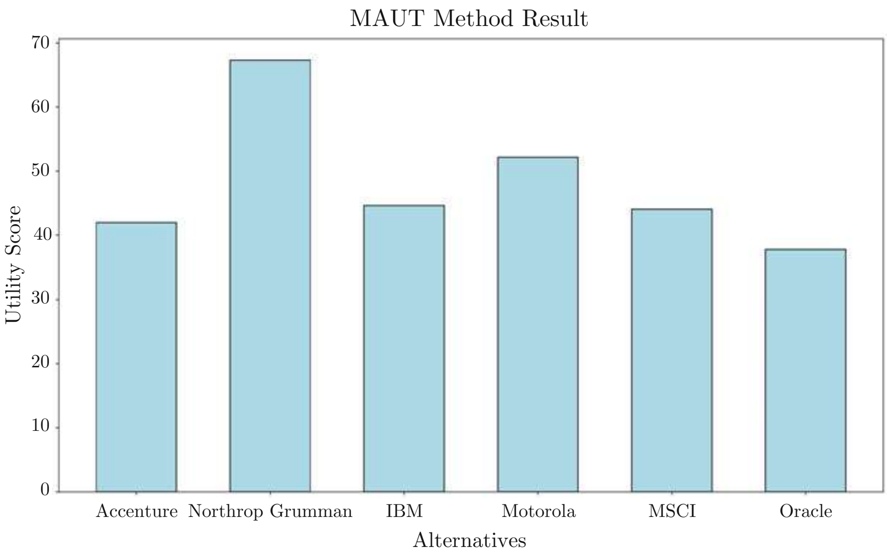
```
                tempSum += normalizedMatrix[i][j] *␣
↪weights[j]
        utilityScore[i] = round(tempSum,4)

df = pd.DataFrame(utilityScore, index=companyName,␣
↪columns=["Score"])
display(df)
```

```
=============== Utility Score ===============
  Company    Score
Accenture   0.4193
Northrop    0.6732
IBM         0.4459
Motorola    0.5215
MSCI        0.4402
Oracle      0.3774
```

Step 3: Finally, the results should be sorted in order to find the ranking of the method. The following code describes this process and offers a visualization of the results. For this purpose, we use the commands of the matplotlib library in order to create a figure and print it.

[6]:
```
tupledList = list(zip(companyName,utilityScorePer))
tupledListSorted = sorted(tupledList, key=lambda tup:␣
↪tup[1], reverse=True)

df = pd.DataFrame(tupledListSorted,␣
↪columns=["Company", "Score"])
df.index = df.index + 1
display(df)

plt.figure(figsize=(16,9))
plt.bar(companyName, utilityScorePer, color =␣
↪'lightblue', edgecolor = 'black', width=0.6)
plt.xlabel(r" Alternatives", fontsize=16)
plt.ylabel(r"Utility Score", fontsize=16)
plt.title(r'MAUT Method Result', fontsize=21)
plt.xticks(fontsize=12, rotation=0)
```

```
=============== MAUT Ranking ===============
            Company  Score
1  Northrop Grumman  67.32
2          Motorola  52.15
3               IBM  44.59
4              MSCI  44.02
5         Accenture  41.93
6            Oracle  37.74
```



6.5.2 TOPSIS

Step 1: In TOPSIS method we initially calculate the normalized decision matrix as you can see in the following cell of code. As mentioned in the previous chapter, TOPSIS follows a different formula for the normalization of the decision matrix compared to MAUT. Therefore, we present the code which is used to calculate the normalized decision matrix below:

```
[7]: normalizedDecisionMatrix = [[0 for i in␣
      ↪range(criteria)] for y in range(alternatives)]
     for j in range(criteria):
         sumOfPows = 0
         for i in range(alternatives):
```

```
        sumOfPows = sumOfPows + math.
↪pow(decisionMatrix[i][j],2)
        sqSumOfPows =  math.sqrt(sumOfPows)
        for i in range(alternatives):
            normalizedDecisionMatrix[i][j] =␣
↪round(decisionMatrix[i][j]*1.0 / sqSumOfPows,3)

df = pd.DataFrame.
↪from_records(normalizedDecisionMatrix,␣
↪index=companyName ,columns=criterionName)
display(df)
```

```
=============== Normalized decision Matrix ===============
  Company    P/E    EPS  Rev(B)   Beta  D.Yield  Techn  YTD(%)  1 Year
Accenture  0.421  0.280  0.419  0.411    0.305  0.421   0.312   0.223
Northrop   0.305  0.771  0.319  0.313    0.252  0.421   0.520   0.320
IBM        0.197  0.457  0.755  0.533    0.805  0.337   0.248   0.003
Motorola   0.527  0.207  0.074  0.231    0.233  0.421   0.491   0.603
MSCI       0.565  0.252  0.014  0.439    0.214  0.421   0.524   0.649
Oracle     0.305  0.114  0.383  0.450    0.309  0.421   0.235   0.250
```

Step 2: After the calculation of the normalized matrix we incorporate the offsets of the criteria, and thus we calculate the weighted decision matrix with the following formula:

[8]:
```
weightedDecisionMatrix = [[0 for i in␣
↪range(criteria)] for y in range(alternatives)]
for j in range(criteria):
    for i in range(alternatives):
        weightedDecisionMatrix[i][j] =␣
↪round(normalizedDecisionMatrix[i][j] * weights[j],3)

df = pd.DataFrame.
↪from_records(weightedDecisionMatrix,␣
↪index=companyName ,columns=criterionName)
display(df)
```

```
=============== Weighted decision Matrix ===============
  Company    P/E    EPS  Rev(B)   Beta  D.Yield  Techn  YTD(%)  1 Year
Accenture  0.053  0.035  0.052  0.051    0.038  0.053   0.039   0.028
Northrop   0.038  0.096  0.040  0.039    0.032  0.053   0.065   0.040
IBM        0.025  0.057  0.094  0.067    0.101  0.042   0.031   0.000
Motorola   0.066  0.026  0.009  0.029    0.029  0.053   0.061   0.075
MSCI       0.071  0.032  0.002  0.055    0.027  0.053   0.066   0.081
Oracle     0.038  0.014  0.048  0.056    0.039  0.053   0.029   0.031
```

Steps 3–4: The following step consists of the calculation of the positive and negative ideal solutions, which can be found if we use the optimal value for each criterion. After finding these solutions we calculate the distance of each alternative from them, which are used as a measure of how well each alternative performs. These calculations, as well as, the results are presented in the following section:

[9]:
```
idealSolution = [0 for i in range(criteria)]
nonIdealSolution = [0 for i in range(criteria)]
for j in range(criteria):
    maxValue = float('-inf')
    minValue = float('inf')
    for i in range(alternatives):
        if weightedDecisionMatrix[i][j] < minValue:
            minValue = weightedDecisionMatrix[i][j]
        if weightedDecisionMatrix[i][j] > maxValue:
            maxValue = weightedDecisionMatrix[i][j]
    if optimizationType[j] == 0:
        idealSolution[j] = maxValue
        nonIdealSolution[j] = minValue
    elif optimizationType[j] == 1:
        idealSolution[j] = minValue
        nonIdealSolution[j] = maxValue

sPlus = [0 for i in range(alternatives)]
sMinus = [0 for i in range(alternatives)]
for i in range(alternatives):
    sumPlusTemp = 0
    sumMinusTemp = 0
    for j in range(criteria):
        sumPlusTemp = sumPlusTemp + math.
↪pow(idealSolution[j]-weightedDecisionMatrix[i][j],2)
        sumMinusTemp = sumMinusTemp + math.
↪pow(nonIdealSolution[j]-weightedDecisionMatrix[i][j],2)
    sPlus[i] = math.sqrt(sumPlusTemp)
    sMinus[i] = math.sqrt(sumMinusTemp)
```

```
============ Positive/Negative Ideal Solution ============
     Criterion  Positive Score  Negative Score
P/E Ratio                0.025           0.071
EPS                      0.096           0.014
Revenue (B)              0.094           0.002
Beta                     0.029           0.067
Dividend Yield           0.101           0.027
Monthly                  0.053           0.042
YTD (%)                  0.066           0.029
1 Year                   0.081           0.000

===== Distance from Positive/Negative Ideal Solutions =====
  Company  Dis. from Positive  Dis. from Negative
Accenture               0.119               0.068
```

```
Northrop                0.098              0.114
IBM                     0.104              0.133
Motorola                0.138              0.091
MSCI                    0.144              0.092
Oracle                  0.132              0.067
```

Step 5: Finally, the last step involves the calculation of the relative closeness of every alternative to the ideal solution, which is presented in the following cell.

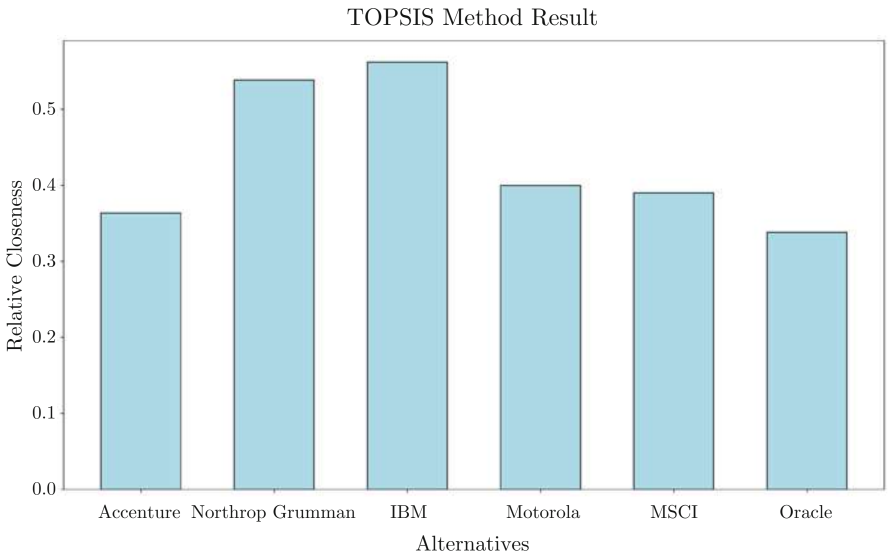
[10]:
```
C = [0 for i in range(alternatives)]
for i in range(alternatives):
    C[i] = sMinus[i]*1.0 / (sMinus[i] + sPlus[i])

df = pd.DataFrame(C, index=companyName,␣
↪columns=["Distance"])
display(df)
```

```
=============== Relative Closeness ===============
  Company   Distance
Accenture      0.363
Northrop       0.538
IBM            0.562
Motorola       0.399
MSCI           0.390
Oracle         0.338
```

The final ranking of TOPSIS method is provided if we sort the relative closeness. In the following cell we describe this procedure as well the visualization of the results.

[11]:
```
tupledList = list(zip(companyName,C))
tupledListSorted = sorted(tupledList, key=lambda tup:␣
↪tup[1], reverse=True)

df = pd.DataFrame(tupledListSorted,␣
↪columns=["Company", "Score"])
df.index = df.index + 1
display(df)

plt.figure(figsize=(16,9))
plt.bar(companyName, C, color = 'lightblue',␣
↪edgecolor = 'black', width=0.6)
plt.xlabel(r" Alternatives", fontsize=16)
plt.ylabel(r"Relative Closeness", fontsize=16)
plt.title(r'TOPSIS Method Result', fontsize=21)
plt.xticks(fontsize=12, rotation=0)
```

```
=============== TOPSIS Ranking ===============

            Company  Score
1               IBM  56.20
2  Northrop Grumman  53.82
3          Motorola  39.95
4              MSCI  39.00
5         Accenture  36.34
6            Oracle  33.80
```



ELECTRE III

Steps 1-2-3: ELECTRE III belongs to the ELECTRE family methods which are based on a different concept than the previous ones. This method is based on pair-wise comparisons between the alternatives. Firstly, we calculate the concordance and discordance tables as described in the following cell of code.

[12]:
```
sumOfWeights = sum(weights)
agreementTable = [[0 for i in range(alternatives)]␣
↪for y in range(alternatives)]
for k in range(criteria):
    if optimizationType[k] == 0:
```

```
        for i in range(alternatives):
            for j in range(alternatives):
                if i!=j:
                    if decisionMatrix[j][k] - decisionMatrix[i][k] <= indifferenceThreshold[k]:
                        agreementTable[i][j] = round(agreementTable[i][j] + 1.0 * weights[k],2)
                    elif decisionMatrix[j][k] - decisionMatrix[i][k] <= preferenceThreshold[k]:
                        agreementTable[i][j] = round(agreementTable[i][j] + ((decisionMatrix[i][k] - decisionMatrix [j][k] + preferenceThreshold[k])*1.0 / (preferenceThreshold[k]-indifferenceThreshold[k])) * weights[k],2)
                    else:
                        agreementTable[i][j] = round(agreementTable[i][j] + 0.0 * weights[k],2)
    elif optimizationType[k] == 1:
        for i in range(alternatives):
            for j in range(alternatives):
                if i!=j:
                    if decisionMatrix[i][k] - decisionMatrix[j][k] <= indifferenceThreshold[k]:
                        agreementTable[i][j] = round(agreementTable[i][j] + 1.0 * weights[k],2)
                    elif decisionMatrix[i][k] - decisionMatrix[j][k] <= preferenceThreshold[k]:
                        agreementTable[i][j] = round(agreementTable[i][j] + ((decisionMatrix[j][k] - decisionMatrix [i][k] + preferenceThreshold[k])*1.0 / (preferenceThreshold[k]-indifferenceThreshold[k])) * weights[k],2)
                    else:
                        agreementTable[i][j] = round(agreementTable[i][j] + 0.0 * weights[k],2)

disagreementTable = [[[0 for k in range(criteria)] for i in range(alternatives)] for j in range(alternatives)]
for k in range(criteria):
    if optimizationType[k] == 0:
```

```
        for i in range(alternatives):
            for j in range(alternatives):
                if i!=j:
                    if decisionMatrix[j][k] -␣
↪decisionMatrix[i][k] <= preferenceThreshold[k]:
                        disagreementTable[i][j][k] = 0
                    elif decisionMatrix[j][k] -␣
↪decisionMatrix[i][k] <= vetoThreshold[k]:
                        disagreementTable[i][j][k] =␣
↪round(((decisionMatrix[j][k] - decisionMatrix␣
↪[i][k] - preferenceThreshold[k])*1.0 /␣
↪(vetoThreshold[k]-preferenceThreshold[k])),2)
                    else:
                        disagreementTable[i][j][k] = 1
    elif optimizationType[k] == 1:
        for i in range(alternatives):
            for j in range(alternatives):
                if i!=j:
                    if decisionMatrix[i][k] -␣
↪decisionMatrix[j][k] <= indifferenceThreshold[k]:
                        disagreementTable[i][j][k] = 0
                    elif decisionMatrix[i][k] -␣
↪decisionMatrix[j][k] <= vetoThreshold[k]:
                        disagreementTable[i][j][k] =␣
↪round(((decisionMatrix[j][k] - decisionMatrix␣
↪[i][k] + preferenceThreshold[k])*1.0 /␣
↪(vetoThreshold[k]-preferenceThreshold[k])),2)
                    else:
                        disagreementTable[i][j][k] = 1
```

Step 4: In this step we calculate the degree of the outranking relationship based on the concordance and discordance indexes that were calculated in the previous steps:

[13]:
```
reliabilityTable = [[0 for i in range(alternatives)]␣
↪for y in range(alternatives)]
for i in range(alternatives):
    for j in range(alternatives):
        if i!=j:
            reliabilityTable[i][j] =␣
↪agreementTable[i][j]
            for k in range(criteria):
                if agreementTable[i][j] <␣
↪disagreementTable[i][j][k]:
```

```
                    reliabilityTable[i][j] =␣
↪round(reliabilityTable[i][j] * ((1 -␣
↪disagreementTable[i][j][k]) / (1 -␣
↪agreementTable[i][j])), 2)

d = 0.8
dominanceTable = [[0 for i in range(alternatives)]␣
↪for y in range(alternatives)]
for i in range(alternatives):
    for j in range(alternatives):
        if i!=j and reliabilityTable[i][j] >= d:
            dominanceTable[i][j] = 1
```

Steps 5-6-7: Finally, the proposed version of ELECTRE III suggests the calculation of the positive and negative flow for each alternative. Based on these two, we can compute the final ELECTRE III net flow, as described in the following cell:

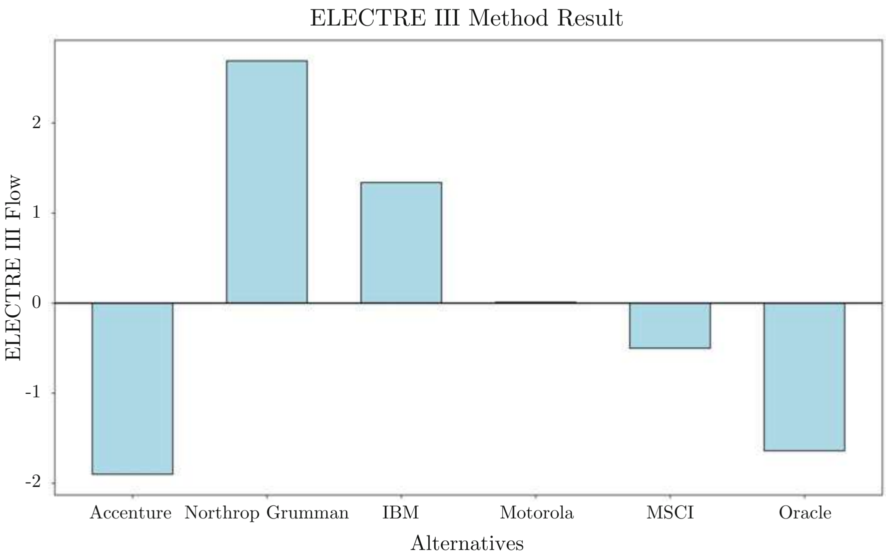
[14]:
```
phiPlus = [round(sum(x),6) for x in reliabilityTable ]
phiMinus = [round(sum(x),6) for x in␣
↪zip(*reliabilityTable)]
phiEl = [round(x1 - x2,6) for (x1, x2) in␣
↪zip(phiPlus, phiMinus)]

df = pd.DataFrame(phiPlus, index=companyName,␣
↪columns=["Flow"])
display(df)
df = pd.DataFrame(phiMinus, index=companyName,␣
↪columns=["Flow"])
display(df)
df = pd.DataFrame(phiEl, index=companyName,␣
↪columns=["Flow"])
display(df)
```

```
=============== Positive/Negative/Net Flow ===============
  Company  Positive Flow  Negative Flow  Net Flow
Accenture           2.31           4.21     -1.90
Northrop            3.59           0.90      2.69
IBM                 1.71           0.37      1.34
Motorola            2.56           2.55      0.01
MSCI                1.63           2.13     -0.50
Oracle              1.77           3.41     -1.64
```

The final ranking of ELECTRE III method is provided sorting the ELECTRE III flows. In the following cell of code, we describe this procedure as well the visualization of the results:

[15]:
```
tupledList = list(zip(companyName,phiEl))
tupledListSorted = sorted(tupledList, key=lambda tup:␣
↪tup[1], reverse=True)

df = pd.DataFrame(tupledListSorted,␣
↪columns=["Company", "Score"])
df.index = df.index + 1
display(df)

plt.figure(figsize=(16,9))
plt.bar(companyName, phiEl, color = 'lightblue',␣
↪edgecolor = 'black', width=0.6)
plt.xlabel(r" Alternatives", fontsize=16)
plt.ylabel(r"ELECTRE III Flow", fontsize=16)
plt.title(r"ELECTRE III Method Result", fontsize=21)
plt.xticks(fontsize=12, rotation=0)
matplotlib.pyplot.axhline(0, c="k")
```

```
=============== ELECTRE III Ranking ===============
            Company  Score
1  Northrop Grumman   2.69
2               IBM   1.34
3          Motorola   0.01
4              MSCI  -0.50
5            Oracle  -1.64
6         Accenture  -1.90
```



6.5.3 PROMETHEE II

Steps 1-2 Finally, PROMETHEE II method is also based on pairwise comparisons of the alternatives. It differs from ELECTRE III as it gives the opportunity to customize the comparison function based on the decision-maker's preferential system. Therefore, we should begin the source code presentation of PROMETHEE II method with the definition of the five available functions for pairwise comparisons. In the following cell of code, we present the *linear preference and indifference function*, which is the most general criterion (the source code for the other criteria can be found on the GitHub repository along with the whole source code). These functions support the required pairwise comparisons between the alternatives and result in the calculation of the preference degrees.

[16]:
```
def
linearPreferenceAndIndifferenceCriterion(evaluationTable,
k, alternatives, decisionMatrix, indifferenceThreshold, preferenceThreshold,
weights, optimizationType):
    if optimizationType[k] == 0:
        for i in range(alternatives):
            for j in range(alternatives):
                if i!=j and decisionMatrix[i][k] >= decisionMatrix[j][k]:

                    if decisionMatrix[i][k]
 - decisionMatrix[j][k] > preferenceThreshold[k]:

                        evaluationTable[i][j] =
 evaluationTable[i][j] + 1.0 * weights[k]
```

```
                    elif decisionMatrix[i][k]
- decisionMatrix[j][k] > indifferenceThreshold[k]:

                        evaluationTable[i][j] =
evaluationTable[i][j] + ((decisionMatrix[i][k]

                        - decisionMatrix [j][k] - indifferenceThreshold[k])*1.0 /
(preferenceThreshold[k]-indifferenceThreshold[k]))
* weights[k]

                    else:

                        evaluationTable[i][j] =
evaluationTable[i][j] + 0.0 * weights[k]
    elif optimizationType[k] == 1:
        for i in range(alternatives):
            for j in range(alternatives):
                if i!=j and decisionMatrix[i][k] >= decisionMatrix[j][k]:
                    if decisionMatrix[i][k] - decisionMatrix[j][k] >␣
↪preferenceThreshold[k]:
                        evaluationTable[j][i] = evaluationTable[j][i] + 1.0 *␣
↪weights[k]
                    elif decisionMatrix[i][k] - decisionMatrix[j][k] >␣
↪indifferenceThreshold[k]:
                        evaluationTable[j][i] = evaluationTable[j][i] +␣
↪((decisionMatrix[i][k] - decisionMatrix [j][k] - indifferenceThreshold[k])*1.0 /
↪ (preferenceThreshold[k]-indifferenceThreshold[k])) * weights[k]
                    else:
                        evaluationTable[j][i] = evaluationTable[j][i] + 0.0 *␣
↪weights[k]
```

Step 3: Having defined the necessary functions we can calculate the multicriteria preference degree between all the alternatives. This is achieved calling the functions that we have developed in the previous steps. The criterion used must be defined by the user in the .csv file in the beginning of the process:

[17]:
```
evaluationTable = [[0.0 for i in range(alternatives)]␣
↪for y in range(alternatives)]

for k in range(criteria):
    if criterion[k] == 1:
        usualCriterion(evaluationTable, k,␣
↪alternatives, decisionMatrix,␣
↪indifferenceThreshold, preferenceThreshold,␣
↪weights, optimizationType)
    elif criterion[k] == 2:
        quasiCriterion(evaluationTable, k,␣
↪alternatives, decisionMatrix,␣
↪indifferenceThreshold, preferenceThreshold,␣
↪weights, optimizationType)
    elif criterion[k] == 3:
```

```
            linearPreferenceCriterion(evaluationTable, k,␣
↪alternatives, decisionMatrix,␣
↪indifferenceThreshold, preferenceThreshold,␣
↪weights, optimizationType)
        elif criterion[k] == 4:
            levelCriterion(evaluationTable, k,␣
↪alternatives, decisionMatrix,␣
↪indifferenceThreshold, preferenceThreshold,␣
↪weights, optimizationType)
        elif criterion[k] == 5:
            ␣
↪linearPreferenceAndIndifferenceCriterion(evaluation␣
↪Table, k, alternatives, decisionMatrix,␣
↪indifferenceThreshold, preferenceThreshold,␣
↪weights, optimizationType)
```

Step 4: After the calculation of the evaluation table, we can compute the positive and negative multicriteria preference flows. The net flow of PROMETHEE method can be calculated if we subtract these two flows:

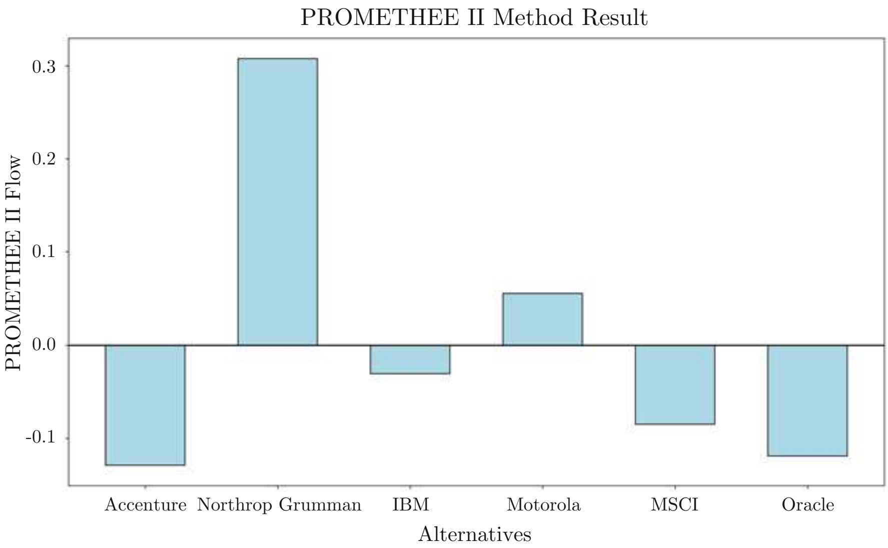
[18]:
```
sumOfLines = np.sum(evaluationTable, axis=1)
sumOfColumns = np.sum(evaluationTable, axis=0)

phiPlus = sumOfLines*1.0 / (alternatives - 1)
phiMinus = sumOfColumns*1.0 / (alternatives - 1)
phi = phiPlus - phiMinus

df = pd.DataFrame(phiPlus, index=companyName,␣
↪columns=["Flow"])
display(df)
df = pd.DataFrame(phiMinus, index=companyName,␣
↪columns=["Flow"])
display(df)
df = pd.DataFrame(phi, index=companyName,␣
↪columns=["Flow"])
display(df)
```

```
=============== Positive/Negative/Net Flow ===============
          Company  Positive Flow  Negative Flow  Net Flow
Accenture                  0.158          0.287    -0.128
Northrop Grumman           0.423          0.116     0.307
IBM                        0.385          0.415    -0.030
Motorola                   0.301          0.245     0.055
MSCI                       0.205          0.289    -0.084
Oracle                     0.163          0.282    -0.119
```

The final ranking of PROMETHEE II method is provided if we sort the net flows. In the following cell of code, we describe this procedure as well the visualization of the results:

```
[19]: tupledList = list(zip(companyName,phiEl))
tupledListSorted = sorted(tupledList, key=lambda tup:␣
 ↪tup[1], reverse=True)

df = pd.DataFrame(tupledListSorted,␣
 ↪columns=["Company", "Score"])
df.index = df.index + 1
display(df)

plt.figure(figsize=(16,9))
plt.bar(companyName, phi, color = 'lightblue',␣
 ↪edgecolor = 'black', width=0.6)
plt.xlabel(r"Alternatives", fontsize=16)
plt.ylabel(r"PROMETHEE II Flow", fontsize=16)
plt.title(r"PROMETHEE II Method Result", fontsize=21)
plt.xticks(fontsize=12, rotation=0)
matplotlib.pyplot.axhline(0, c="k")
```

```
=============== PROMETHEE II Ranking ===============
            Company  Score
1  Northrop Grumman   2.69
2               IBM   1.34
3          Motorola   0.01
4              MSCI  -0.50
5            Oracle  -1.64
6         Accenture  -1.90
```



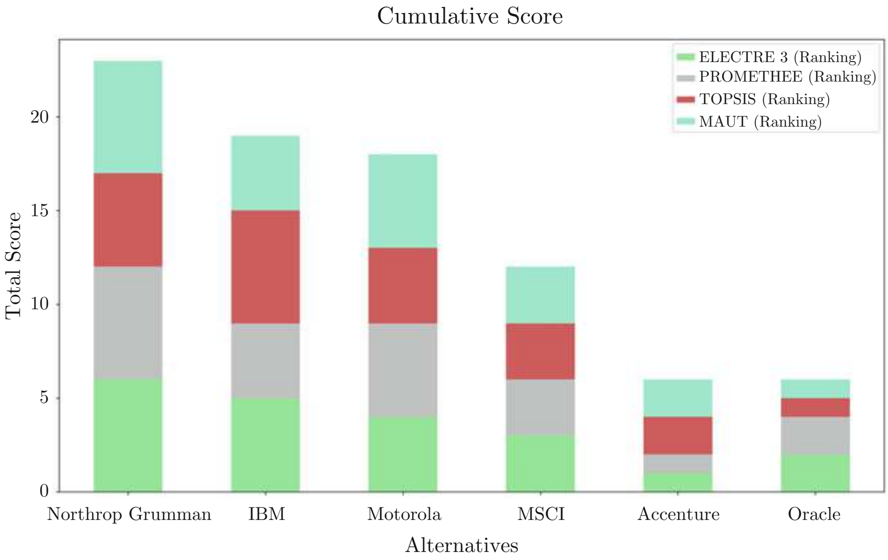
6.5.4 Cumulative Ranking

The final ranking is the weighted sum of the rankings of the four MCDA methods. Therefore, in the following cell we describe the calculation of the cumulative score of each alternative as well as the final ranking and a visualization of the results.

[20]:
```
decisionMatrix = np.array(decisionMatrix)
data = {'Name':companyName, 'ELECTRE III':phiEl, 'MAUT':
↪utilityScore, 'PROMETHEE II':phi, 'TOPSIS':C}

df = pd.DataFrame(data)
df["ELECTRE 3 (Ranking)"] = df["ELECTRE 3"].rank()
df["MAUT (Ranking)"] = df["MAUT"].rank()
df["PROMETHEE (Ranking)"] = df["PROMETHEE"].rank()
df["TOPSIS (Ranking)"] = df["TOPSIS"].rank()
df["RankSum"] = df.iloc[:,13:17].sum(axis=1)
df.sort_values("RankSum", inplace=True, ascending=False)
display(df)

ax = df.iloc[:50].plot(x ='Name',
                 y={'ELECTRE 3 (Ranking)','MAUT (Ranking)','TOPSIS␣
↪(Ranking)','PROMETHEE (Ranking)'},
             color={'lightgreen','silver','aquamarine','indianred'},
             kind = 'bar',
             figsize=(16, 9))
legend = plt.legend()
plt.setp(legend.get_texts(), color='grey', size='16')
plt.xlabel(r"Alternatives", fontsize=16)
```

```
plt.ylabel(r"Total Score", fontsize=16)
plt.title(r"Cumulative Score", fontsize=21)
plt.xticks(fontsize=12, rotation=0)
matplotlib.pyplot.axhline(0, c="k")
```

```
====================  Final Ranking  ====================
       Name  ELECTRE III  MAUT  PROMETHEE II  TOPSIS  RankSum
1  Northrop            1     1             1       2     23.0
2       IBM            2     3             3       1     19.0
3  Motorola            3     2             2       3     18.0
4      MSCI            4     4             4       4     12.0
5 Accenture            6     5             6       5      6.0
6    Oracle            5     6             5       6      6.0
```



6.5.5 Multiobjective Portfolio Optimization

In this part, we present the implementation of the second phase of the methodology, which is multiobjective portfolio optimization. In the beginning, we describe the connection with the Yahoo Finance API. Subsequently, we show some fundamental financial calculations with Python, such as stock return and statistical indexes of the securities. Finally, we make a quick presentation of the Python solvers that were used for portfolio optimization.

Connection to the Yahoo API

After importing all necessary libraries, we must define the securities, as well as the starting and ending date of the empirical testing. The list tickers contains the names of the securities and the variables startDate, endDate contain the duration of the simulation. As you can see, the selected time horizon is 3 years, from 1-1-2016 until 31-12-2018 and the selected equities of this example are 6 securities from the technology sector of NYSE stock exchange.

[1]:
```
tickers = ['ACN', 'NOC', 'IBM', 'MSI', 'MSCI', 'ORA' ]

startDate = '2016-01-01'
endDate = '2018-12-31'
```

In the following cell of code, we describe the connection to the Yahoo Finance API. This connection is made with the data.DataReader function of the pandas library, which takes four arguments: (a) a list with the names of the securities, (b) the starting date, (c) the ending date, and (d) the name of the API (in this case "yahoo"). For example, in the following cell you can see the historical values of the first security "ACN." The result contains the values "High," "Low," "Open," "Close," and "Volume."

[2]:
```
historicalDataNOC = data.DataReader('ACN', 'yahoo',␣
↪startDate, endDate)
display(historicalDataNOC)
```

```
===========  Raw Data from Yahoo API  ===========
                  High         Low        Open       Close     Volume
Date
2016-01-04  102.650002  100.970001  102.620003  101.830002  2817000.0
2016-01-05  102.870003  101.470001  101.970001  102.360001  2409000.0
2016-01-06  103.059998  100.540001  100.809998  102.160004  3134200.0
2016-01-07  100.839996   98.839996   99.750000   99.160004  3194700.0
2016-01-08   99.809998   98.000000   99.480003   98.199997  2330200.0
...                ...         ...         ...         ...        ...
2018-12-31  141.289993  139.389999  140.399994  141.009995  1826400.0
```

From all the provided security values, we only need the data of the "Open" column in order to use it for the empirical testing. Therefore, we firstly draw all the desired data from the Yahoo API and then we discard the unnecessary columns, as shown in the following section:

[3]:
```
historicalValues = data.DataReader(tickers, 'yahoo',␣
↪startDate, endDate)
stockValues = historicalValues['Open']
numOfDates = stockValues.shape[0]
numOfSecurities = stockValues.shape[1]
print("Number of securities:", numOfSecurities)
```

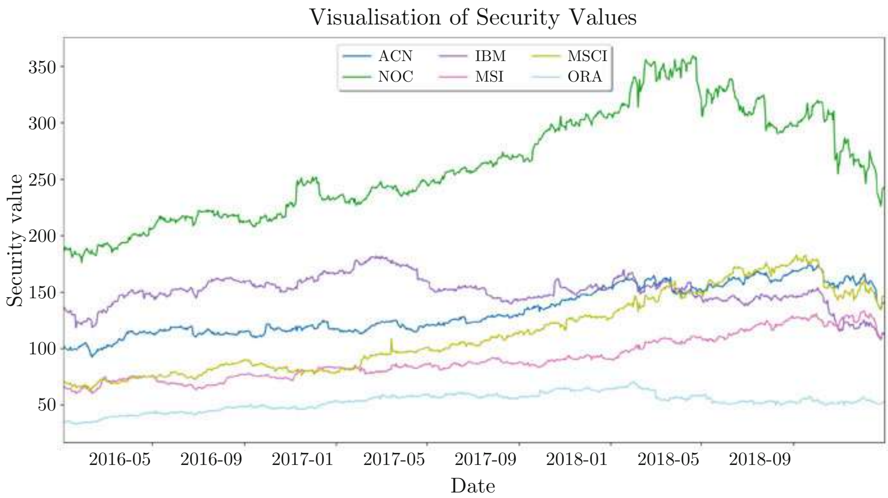
```
print("Number of dates:", numOfDates, "\n")
stockValues = stockValues.fillna(method='ffill')
display(stockValues)
```

```
Number of securities: 6
Number of dates: 754

==============  Stock Values  ==============
Symbols         ACN       NOC        IBM       MSI      MSCI       ORA
Date
2016-01-04 102.6200 185.9799  135.6000   67.6699   71.0999   35.6300
2016-01-05 101.9700 187.8500  136.7599   66.5000   70.6100   35.1500
2016-01-06 100.8099 190.1600  134.3800   65.5700   70.2600   34.7900
2016-01-07 99.7500  187.8999  133.6999   64.6299   69.2399   35.0400
2016-01-08 99.4800  188.7899  133.1799   64.3300   69.5800   35.5200
...             ...       ...       ...       ...       ...       ..
↪.
2018-12-31 140.3999 243.3099 113.3300 113.0299 146.5000  53.0099
```

Security Values and Security Returns

In this point, the dataframe stockValues contains the historical values of the securities. These values can be easily visualized with matplotlib library in a common diagram with the following piece of code:

```
[4]: fig = stockValues.plot(figsize=(16,9), cmap='tab20')
plt.xlabel(r"Date", fontsize=16)
plt.ylabel(r"Security value", fontsize=16)
plt.title(r"Visualization of Security Values",␣
 ↪fontsize=21)
plt.xticks(fontsize=12, rotation=0)
plt.legend(loc='upper center', bbox_to_anchor=(0.5,␣
 ↪1), ncol=3, fancybox=True, shadow=True)
```



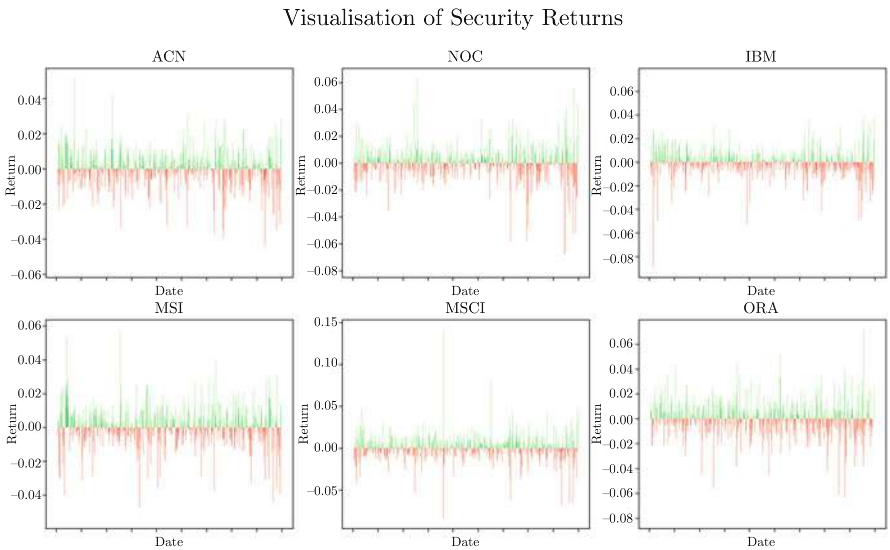
The following step is the calculation of the arithmetical return of the securities. This step is executed by converting the pandas dataframe to NumPy array in order to make the calculations and then converting the returns list back to dataframe. Given the historical values the calculation of the arithmetical return is presented below:

[5]:
```
stockValuesArray = pd.DataFrame(stockValues).
↪to_numpy()
stockReturnsArray = np.empty(shape = (numOfDates-1,␣
↪numOfSecurities))
for i in range(numOfSecurities):
    for j in range(numOfDates-1):
        stockReturnsArray[j][i] =␣
↪(stockValuesArray[j+1][i]-stockValuesArray[j][i])/
↪stockValuesArray[j][i]
returnDates = stockValues.index[1:]
stockReturns = pd.DataFrame(stockReturnsArray,␣
↪index=returnDates, columns=stockValues.columns)
display(stockReturns)
```

```
============  Stock Returns  ============
Symbols          ACN       NOC       IBM       MSI      MSCI       ORA
Date
2016-01-05 -0.006334  0.010055  0.008554 -0.017290 -0.006892 -0.013472
2016-01-06 -0.011376  0.012297 -0.017403 -0.013985 -0.004957 -0.010242
2016-01-07 -0.010515 -0.011885 -0.005060 -0.014336 -0.014518  0.007186
2016-01-08 -0.002707  0.004737 -0.003889 -0.004642  0.004911  0.013699
2016-01-11 -0.010052 -0.006939 -0.010287  0.005130 -0.011785  0.006475
...              ...       ...       ...       ...       ...       ...
2018-12-31 -0.004326  0.008037 -0.007792  0.004622  0.001025  0.009907
```

We can also create a common diagram with the securities' returns. The diagram of the return of each security can be created with the following cell of code and the corresponding diagram is presented in the following figure:

```
[6]: fig, ax = plt.subplots(2, 3, figsize=(16,9))
     for j in range(numOfSecurities):
         colors = np.array([(1,0,0)]*len(returnDates))
         for i in range(numOfDates-1):
             if stockReturnsArray[i][j] > 0:
                 colors[i] = (0,1,0)
         ax[j//3,j%3].bar(returnDates, stockReturnsArray[:
     ↪,j], color=colors)
         ax[j//3,j%3].set_title(stockValues.columns[j])
         ax[j//3,j%3].set_xticklabels([])
         ax[j//3,j%3].set_xlabel("Date")
         ax[j//3,j%3].set_ylabel("Return")
         fig.suptitle('Visualization of Security Returns',␣
     ↪fontweight='bold', fontsize=21)
```



Financial Statistics

In this step, we calculate a series of fundamental statistical indexes for the selected securities. These calculations are made with the NumPy Library which supports a variety of statistical calculations. Additionally, we have used two functions from the *scipy.stats* library in order to calculate the skewness and the kurtosis of the securities, as shown in the following section:

[7]:
```
from scipy.stats import kurtosis, skew

for i in range(numOfSecurities):
        MinReturn[i] = np.min(stockReturnsArray[:,i])
        MaxReturn[i] = np.max(stockReturnsArray[:,i])
        MedianReturn[i] = np.
↪median(stockReturnsArray[:,i])
        MeanReturn[i] = np.mean(stockReturnsArray[:
↪,i])
        SD[i] = np.std(stockReturnsArray[:,i])
        VaR99[i] = np.percentile(stockReturnsArray[:
↪,i], 1)
        VaR97[i] = np.percentile(stockReturnsArray[:
↪,i], 3)
        VaR95[i] = np.percentile(stockReturnsArray[:
↪,i], 5)
        Skewness[i] = skew(stockReturnsArray[:,i],␣
↪bias=False)
        Kurtosis[i] = kurtosis(stockReturnsArray[:
↪,i], bias=False)
        AbsMinPerSD[i] = np.abs(MinReturn[i])/SD[i]

statistics = pd.DataFrame(
    {'MinReturn': MinReturn,
     'MaxReturn': MaxReturn,
     'Median': MedianReturn,
     'Mean': MeanReturn,
     'SD': SD,
     'VaR99': VaR99,
     'VaR97': VaR97,
     'VaR95': VaR95,
     'Skewness': Skewness,
     'Kurtosis': Kurtosis,
     'AbsMinPerSD': AbsMinPerSD,
    }, index=stockValues.columns)


display(statistics)
```

```
        MinReturn  MaxReturn   Median     Mean        SD     VaR99
Symbols
ACN     -0.056337   0.051843  0.001509  0.000480  0.011213 -0.035905
```

```
NOC     -0.077808   0.063431  0.000676  0.000448  0.013467 -0.049242
IBM     -0.089540   0.071541  0.000000 -0.000162  0.012330 -0.036385
MSI     -0.054133   0.058159  0.001211  0.000757  0.012276 -0.035977
MSCI    -0.084600   0.142090  0.001597  0.001079  0.015456 -0.039073
ORA     -0.080762   0.072031  0.000496  0.000634  0.014553 -0.042588

            VaR97     VaR95  Skewness   Kurtosis  AbsMinPerSD
Symbols
ACN     -0.022927 -0.017429 -0.658652   3.198245     5.024272
NOC     -0.024005 -0.019069 -0.674012   6.010290     5.777613
IBM     -0.022937 -0.017667 -0.478771   7.086411     7.261800
MSI     -0.026263 -0.018172 -0.169242   2.672691     4.409556
MSCI    -0.028230 -0.021382  0.579821  12.724204     5.473484
ORA     -0.026683 -0.021765 -0.436693   3.334074     5.549322
```

Besides, given the arithmetical returns of the securities we can calculate the covariance matrix of the securities. The computation can be achieved with the pandas function *cov* which calculates the covariance matrix of a pandas dataframe.

```
[8]: cov = stockReturns.cov()
covarianceMatrix = np.array(cov)
display(cov)
```

```
============  Covariance Matrix  ============
Symbols       ACN       NOC       IBM       MSI      MSCI       ORA
ACN      0.000126  0.000056  0.000061  0.000057  0.000078  0.000051
NOC      0.000056  0.000182  0.000045  0.000053  0.000071  0.000035
IBM      0.000061  0.000045  0.000152  0.000050  0.000066  0.000041
MSI      0.000057  0.000053  0.000050  0.000151  0.000082  0.000046
MSCI     0.000078  0.000071  0.000066  0.000082  0.000239  0.000046
ORA      0.000051  0.000035  0.000041  0.000046  0.000046  0.000212
```

Mean–Variance Model

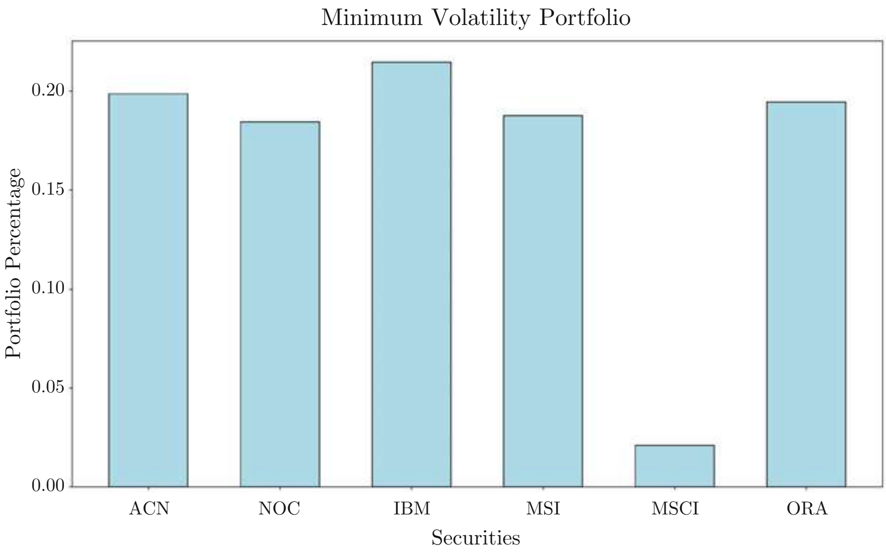
In this section we attempt to optimize the portfolio of the selected securities, using the mean–variance method. The optimization problem is a quadratic bi-objective problem, which will be solved parametrically setting the expected return as a parameter. We use the *scipy.optimizer* library in order to solve the problem.

Initially, we compute the global minimum variance portfolio (GMVP) using the *scipy.minimize* function. The minimize function takes as an argument the mean return column vector and the covariance 2D matrix and computes the proportions of the GMVP minimizing the quantity defined in the function named "Portfolio Volatility" which is the standard deviation of the portfolio. Additionally, we set the constraint that the weights sum to 1 and that the bounds of the proportions are (0,1), imposing the short sales restriction.

```
[9]: #Objective Function
def portfolioVolatility(weights, MeanReturn,␣
↪covarianceMatrix):
    std = np.sqrt(np.dot(weights.T, np.
↪dot(covarianceMatrix, weights)))
    return std

#Constraints
args = (MeanReturn, covarianceMatrix)
constraints = ({'type': 'eq', 'fun': lambda x: np.
↪sum(x) - 1})
bound = (0,1)
bounds = tuple(bound for asset in␣
↪range(numOfSecurities))

#Optimization Function
minVolatilityPortfolio = sco.
↪minimize(portfolioVolatility, numOfSecurities*[1./
↪numOfSecurities,], args=args, method='SLSQP',␣
↪bounds=bounds, constraints=constraints)

sdPort1 = np.sqrt(np.dot(minVolatilityPortfolio['x'].
↪T, np.dot(covarianceMatrix,␣
↪minVolatilityPortfolio['x'])))
retPort1 = np.
↪sum(MeanReturn*minVolatilityPortfolio['x'] )


plt.figure(figsize=(16,9))
plt.bar(tickers, minVolatilityPortfolio['x'], color =␣
↪'lightblue', edgecolor = 'black', width=0.6)
plt.xlabel(r"Securities", fontsize=16)
plt.ylabel(r"Portfolio Percentage", fontsize=16)
plt.title(r"Minimum Volatility Portfolio",␣
↪fontsize=21)
plt.xticks(fontsize=12, rotation=0)
plt.savefig("barplot8.png", dpi=600)
```

```
Risk of minimum volatility portfolio:
0.008501272989289266
Return of minimum volatility portfolio:
0.0004306962318125164 5
```

```
Sharpe Ratio of minimum volatility portfolio:
0.05066255751993256
```



Subsequently, we parametrically solve the same problem in order to gradually find the efficient frontier. Therefore, we compute the efficient portfolios between the GMVP and the maximum return portfolio. As you can see, we have set the variable `numOfPortfolios` equal to 30, in order to calculate 30 Pareto optimal portfolios.

[10]:
```
numOfPortfolios = 30
maxReturn = max(MeanReturn)
returnRange = np.linspace(retPort1, maxReturn,␣
↪numOfPortfolios)
efficientFrontier = []
AllReturns = []
AllSDs = []
for target in returnRange:
    args = (MeanReturn, covarianceMatrix)
    constraints = ({'type': 'eq', 'fun': lambda x: np.
↪sum(MeanReturn*x) - target},
                            {'type': 'eq', 'fun':␣
↪lambda x: np.sum(x) - 1})
    bounds = tuple((0,1) for asset in␣
↪range(numOfSecurities))
```

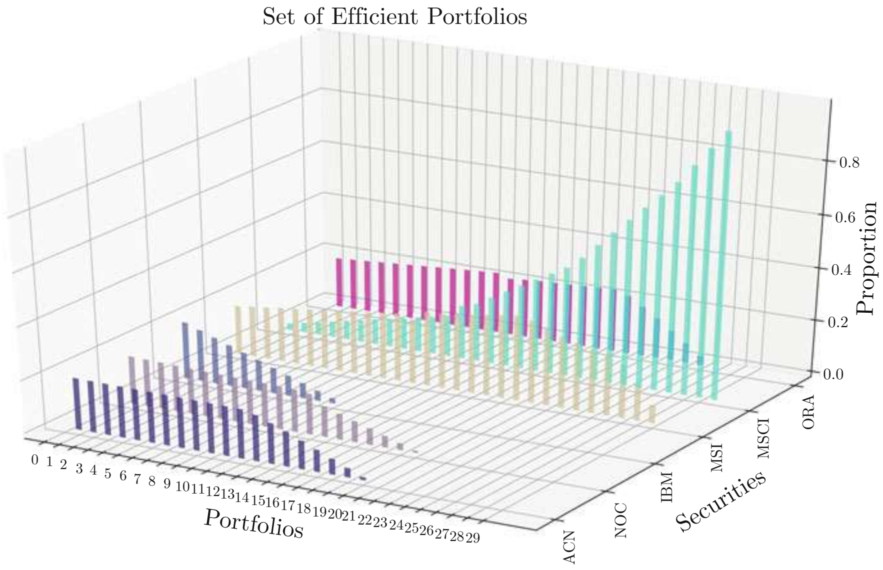
```
    result = sco.minimize(portfolioVolatility,␣
↪numOfSecurities*[1./numOfSecurities,], args=args,␣
↪method='SLSQP', bounds=bounds,␣
↪constraints=constraints)
    efficientFrontier.append(result)
    AllSDs.append(np.sqrt(np.dot(result['x'].T, np.
↪dot(covarianceMatrix, result['x']))))
    AllReturns.append(np.sum(MeanReturn*result['x']))
```

Now, we have calculated a series of Pareto optimal portfolios, as well as their return and standard deviation, also known as portfolio volatility. More specifically, the list `AllReturns` includes the portfolios' return, the list `AllSDs` includes the portfolios' volatility and the 2-dimensional list `efficientFrontier` includes the proportion of each security in each portfolio. Finally, we present the efficient portfolios, displaying them as a dataframe and also plotting a 3D-figure of the portfolios proportions.

[11]:
```
efficientPortfolios = [0 for i in␣
↪range(numOfPortfolios)]
for i in range(numOfPortfolios):
    efficientPortfolios[i] = np.
↪round(efficientFrontier[i].x,2)

df = pd.DataFrame(efficientPortfolios,␣
↪columns=tickers)
df.index = df.index + 1
display(df)

weightingFactor = [[0 for i in␣
↪range(numOfPortfolios)] for j in␣
↪range(numOfSecurities)]
for i in range(numOfSecurities):
    for j in range(numOfPortfolios):
        weightingFactor[i][j] = efficientFrontier[j].
↪x[i]

from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(20,12))
ax = fig.add_subplot(111, projection='3d')
for z in range(numOfSecurities):
    xs = np.arange(numOfPortfolios)
    ys = weightingFactor[z]
```

```
    cs=(random.uniform(0, 1), random.uniform(0, 1),␣
↪random.uniform(0, 1))
    ax.bar(xs, ys, zs=z, zdir='y', color=cs, alpha=0.
↪7, width=0.4)
plt.yticks(np.arange(numOfSecurities), tickers)
plt.xticks(np.arange(numOfPortfolios))
ax.set_xlabel(r'Portfolios', fontsize=16)
ax.set_ylabel(r'Securities', fontsize=16)
ax.set_zlabel(r'Proportion', fontsize=16)
plt.title(r"Set of Efficient Portfolios", fontsize=21)
plt.savefig("barplot11.png", dpi=600)
```

```
     ACN   NOC   IBM   MSI  MSCI   ORA
1   0.20  0.18  0.22  0.19  0.02  0.19
2   0.20  0.18  0.20  0.19  0.04  0.20
3   0.20  0.18  0.18  0.20  0.05  0.20
4   0.19  0.18  0.16  0.21  0.06  0.20
5   0.19  0.17  0.14  0.21  0.07  0.21
6   0.19  0.17  0.12  0.22  0.08  0.21
7   0.19  0.17  0.10  0.23  0.10  0.21
8   0.19  0.17  0.08  0.23  0.11  0.22
9   0.19  0.17  0.06  0.24  0.12  0.22
10  0.19  0.17  0.04  0.25  0.13  0.23
..  ....  ....  ....  ....  ....  ....
30  0.00  0.00  0.00  0.00  1.00  0.00
```



Goal Programming Methodology

In this section we present the goal programming model construction, as well as the python solver used to provide the optimal portfolio. Firstly, the input of the goal programming methodology includes the following variables:

- numSecurities (integer) is the number of securities,
- companyName (str list) contains the names of the securities,
- betaIndex (float list) contains the beta index for each security,
- DYIndex (float list) contains the dividend yield for each security.

[1]:
```
numSecurities = 6
companyName = ['ACN', 'NOC', 'IBM', 'MSI', 'MSCI',␣
↪'ORA']
betaIndex = [0.8, 1.36, 0.59, 1.12, 1.05, 1.15]
Rev = [32.89, 77.86, 7.63, 1.48, 43.22, 39.53]
DYIndex = [1.44, 4.59, 1.33, 1.22, 1.74, 1.76]
```

Secondly, the parameters of the goal programming model are defined below:

- numPortfolios : Number of portfolios to be constructed.
- minSecurities : Minimum number of securities to participate in each portfolio.
- maxSecurities : Maximum number of securities to participate in each portfolio.
- lowerBound : Minimum value of the weight of each security.
- upperBound : Maximum value of the weight of each security.

- capitalThreshold : Threshold that determines the Billions needed to consider a security as a high capitalization investment.

Finally, the target values of the goal programming model are the following:

- betaGoal : The target value for portfolio beta.
- DYGoal : The target value for portfolio dividend yield.
- highCapGoal : The target value for the percentage of high capitalization securities participating in the portfolio.

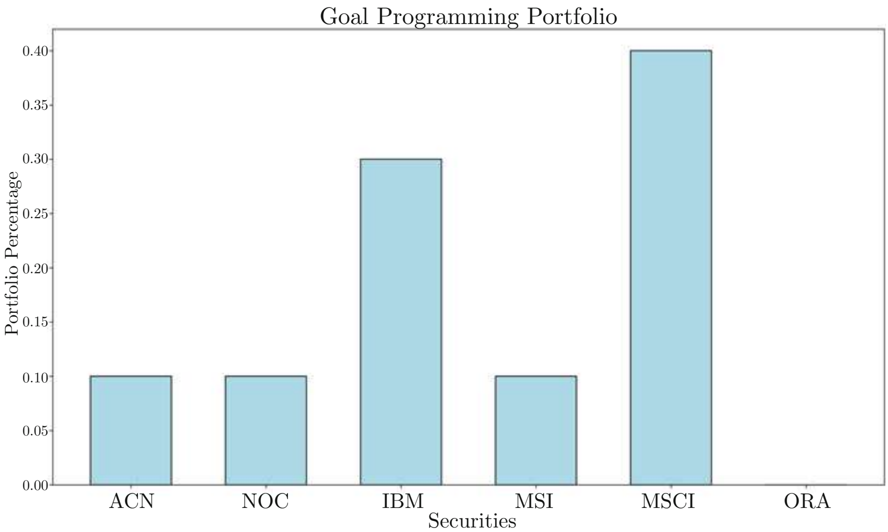
Therefore, the model is constructed as shown in the following cell of code, according to the parameters and the target objectives that we defined above:

[2]:
```
minSecurities = 5
maxSecurities = 6
lowerBound = 0.1
upperBound = 0.5
betaGoal = 0.8
DYGoal = 1.7
highCapGoal = 0.5
capitalThreshold = 40

m = Model()

high = [0 for i in range(numSecurities)]
for i in range(numSecurities):
    if Rev[i] > capitalThreshold:
        high[i] =1

onoff = [ m.add_var(var_type=BINARY) for i in␣
 ↪range(numSecurities) ]
weights = [ m.add_var(var_type=CONTINUOUS) for i in␣
 ↪range(numSecurities) ]

d1P = m.add_var(var_type=CONTINUOUS)
d1M = m.add_var(var_type=CONTINUOUS)
d2P = m.add_var(var_type=CONTINUOUS)
d2M = m.add_var(var_type=CONTINUOUS)
d3P = m.add_var(var_type=CONTINUOUS)
d3M = m.add_var(var_type=CONTINUOUS)

w1P = 1
w1M = 1
w2P = 1
w2M = 1
w3P = 1
```

```python
w3M = 1

m += xsum(weights[i] for i in range(numSecurities)) == 1
m += xsum(onoff[i] for i in range(numSecurities)) <= maxSecurities
m += xsum(onoff[i] for i in range(numSecurities)) >= minSecurities
for i in range(numSecurities):
    m += weights[i] - lowerBound * onoff[i] >= 0
    m += weights[i] - upperBound * onoff[i] <= 0

m += xsum(weights[i] * betaIndex[i] for i in range(numSecurities)) + d1M - d1P == betaGoal
m += xsum(weights[i] * DYIndex[i] for i in range(numSecurities)) + d2M - d2P == DYGoal
m += xsum(weights[i] * high[i] for i in range(numSecurities)) + d3M - d3P == highCapGoal

m.objective = minimize((w1P * d1P + w1M * d1M) / betaGoal + (w2P * d2P + w2M * d2M) / DYGoal + (w3P * d3P + w3M * d3M) / highCapGoal)

status = m.optimize()

portfolio = [0 for i in range(numSecurities)]
for i in range(numSecurities):
    portfolio[i] = onoff[i].x * weights[i].x

plt.figure(figsize=(16,9))
plt.bar(companyName, portfolio, color = 'lightblue', edgecolor = 'black', width=0.6)
plt.xlabel(r"Securities", fontsize=16)
plt.ylabel(r"Portfolio Percentage", fontsize=16)
plt.title(r"Goal Programming Portfolio", fontsize=21)
plt.xticks(fontsize=12, rotation=0)
```

```
=================  Model output  =================
Solution status :  OptimizationStatus.OPTIMAL

Objective function =  0.22683

ACN :  0.1
```

```
NOC :  0.1
IBM :  0.3
MSI :  0.1
MSCI :  0.4
ORA :  0.0

Portfolio beta =  0.925
Portfolio dividend yield =  1.82
High capitalization percentage 0.5
```



Genetic Algorithm Model

As described in Chapter 6.5, the Python solver that we have used for the genetic algorithm model is the *scipy's* function called *differential evolution*. Therefore, we must import the specific library in order to use this function for the solution of the portfolio optimization problem.

[1]:
```
from scipy.optimize import differential_evolution
from scipy.optimize import LinearConstraint, minimize
```

The input data of the genetic algorithm model are values of the securities. The next step involves the calculation of the arithmetic return for the selected securities. However, we have already shown how to calculate the returns of the selected securities, therefore we bypass this step.

We also must calculate the return for the market index, in order to gather all the necessary input for the model. In this case we select the NASDAQ Composite index (ˆIXIC) because the selected securities belong to the NASDAQ stock exchange. Therefore, we calculate the returns of the market index for the selected time horizon.

[2]:
```
historicalValues = data.DataReader('^FCHI', 'yahoo',␣
↪startDate, endDate)
indexValues = historicalValues['Open']
numOfDates = stockValues.shape[0]
indexValues = indexValues.fillna(method='ffill')
display(indexValues)

numPeriods = numOfDates - 1
indexValuesArray = pd.DataFrame(indexValues).
↪to_numpy()
marketReturns = np.empty(shape = (numPeriods))
for j in range(numPeriods):
    marketReturns[j] =␣
↪(indexValuesArray[j+1]-indexValuesArray[j])/
↪indexValuesArray[j]
indexReturns = pd.DataFrame(marketReturns,␣
↪index=returnDates)
```

In this point, we define the multiobjective function `losses`, which is about to be optimized. This function calculates the number of times that the portfolio return is smaller than the selected market index return for all dates during the selected time period. This function needs to be minimized, as the target is to maximize the number of times that the portfolio offers a better return.

[3]:
```
def losses(x):
    losingTimes = 0
    portfReturn = [0 for i in range(numPeriods)]
    for i in range(numPeriods):
        for j in range(numSecurities):
            portfReturn[i] += x[j] * secReturns[i][j]
        if portfReturn[i] < marketReturns[i]:
            losingTimes += 1
    return losingTimes
```

Finally, the function differential_evolution solves this evolutionary problem. The list of variable bounds set the limits for each security's percentage in the portfolio. The function LinearConstraint imposes the capital completeness constraint. The parameters of the model are the following (according to the scipy documentation):

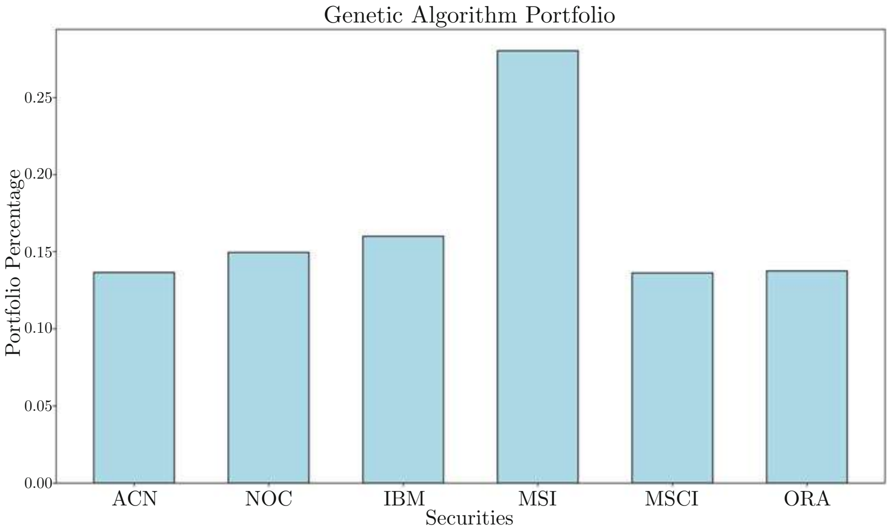
- maxiter: The maximum number of generations over which the entire population is evolved.

- popsize: A multiplier for setting the total population size. The population has popsize * len(x) individuals.
- mutation: The mutation constant. In the literature this is also known as differential weight, being denoted by F.
- recombination: The recombination constant should be in the range [0, 1]. Also known as the crossover probability, being denoted by CR. Increasing this value allows a larger number of mutants to progress into the next generation, but at the risk of population stability.

In the following cell of code, we call the `differential_evolution` function, giving it as an argument the `losses` function in order to calculate the efficient portfolio.

[4]:
```
bounds = [(0,1) for i in range(numSecurities)]
constraints = LinearConstraint(np.
 ↪ones(numSecurities), 1, 1)
result = differential_evolution(losses, bounds,␣
 ↪maxiter=10, polish=True, popsize=30, mutation=1.95,
                                 updating='deferred',␣
 ↪recombination=0.9, constraints=constraints)

for i in range(numSecurities):
    print(companyName[i],": ", np.round(result.
 ↪x[i],4))
print("Percentage that portfolio beats the index:",␣
 ↪(numPeriods-result.fun)/numPeriods)

plt.figure(figsize=(16,9))
plt.bar(companyName, result.x, color = 'lightblue',␣
 ↪edgecolor = 'black', width=0.6)
plt.xlabel(r"Securities", fontsize=16)
plt.ylabel(r"Portfolio Percentage", fontsize=16)
plt.title(r"Genetic Algorithm Portfolio", fontsize=21)
plt.xticks(fontsize=12, rotation=0)
```

```
ACN :  0.1366
NOC :  0.1495
IBM :  0.16
MSI :  0.2802
MSCI :  0.1362
ORA :  0.1375
Percentage that portfolio beats the index: 0.
 ↪5126162018592297
```



MOIP PROMETHEE Flow 3-Dimensional Model

Finally, in this paragraph we present the solver for the 3-Dimensional problem involving the PROMETHEE flow. The 3 objective functions will be the portfolio beta index, the portfolio dividend yield, and the PROMETHEE flow. The input data for this method are the beta indexes, the PROMETHEE net flows, and the dividend yield of each security:

[1]:
```
numSecurities = 6
companyName = ['ACN', 'NOC', 'IBM', 'MSI', 'MSCI',␣
↪'ORA']
betaIndex = [0.8, 1.36, 0.59, 1.12, 1.05, 1.15]
Rev = [32.89, 77.86, 7.63, 1.48, 43.22, 39.53]
DYIndex = [1.44, 4.59, 1.33, 1.22, 1.74, 1.76]
promIndex = [0.3078, -0.0306, 0.0558, -0.0849, -0.
↪1289, -0.1191]
```

The PROMETHEE flow model includes the following parameters, which are used in order to impose the required policy restrictions:

- minSecurities : Minimum number of securities to participate in each portfolio.
- maxSecurities : Maximum number of securities to participate in each portfolio.
- lowerBound : Minimum value of the weight of each security.
- upperBound : Maximum value of the weight of each security.

[2]:
```
minSecurities = 6
maxSecurities = 10
lowerBound = 0.05
upperBound = 0.3
```

We solve the 1-objective optimization problem for each one of the objective functions, in order to find their target values. Firstly, we solve the problem of minimizing the portfolio beta.

[3]:
```
m = Model()

onoff = [ m.add_var(var_type=BINARY) for i in␣
↪range(numSecurities) ]
weights = [ m.add_var(var_type=CONTINUOUS) for i in␣
↪range(numSecurities) ]

m += xsum(weights[i] for i in range(numSecurities))␣
↪== 1
m += xsum(onoff[i] for i in range(numSecurities)) <=␣
↪maxSecurities
m += xsum(onoff[i] for i in range(numSecurities)) >=␣
↪minSecurities
for i in range(numSecurities):
    m += weights[i] - lowerBound * onoff[i] >= 0
    m += weights[i] - upperBound * onoff[i] <= 0

m.objective = minimize(xsum(weights[i] * betaIndex[i]␣
↪for i in range(numSecurities)))

status = m.optimize()

print(status, "\n")
minBeta = m.objective_value
print("Minimum Beta = ", minBeta, "\n")
```

```
OptimizationStatus.OPTIMAL

Minimum Beta =  0.861
```

With the same procedure we solve the maximization problem of the PROMETHEE flow function and finally we solve the problem of maximizing the portfolio dividend yield. The source code for these two objectives functions is exactly the same with the portfolio beta that we explained above, except that we now change the objective function, as shown in the following cell of code.

[4]:
```
m.objective = maximize(xsum(weights[i] * promIndex[i]␣
↪for i in range(numSecurities)))
```

```
OptimizationStatus.OPTIMAL

Maximum PROMETHEE flow =  0.084785
```

[5]:
```
m.objective = maximize(xsum(weights[i] * DYIndex[i]␣
↪for i in range(numSecurities)))
```

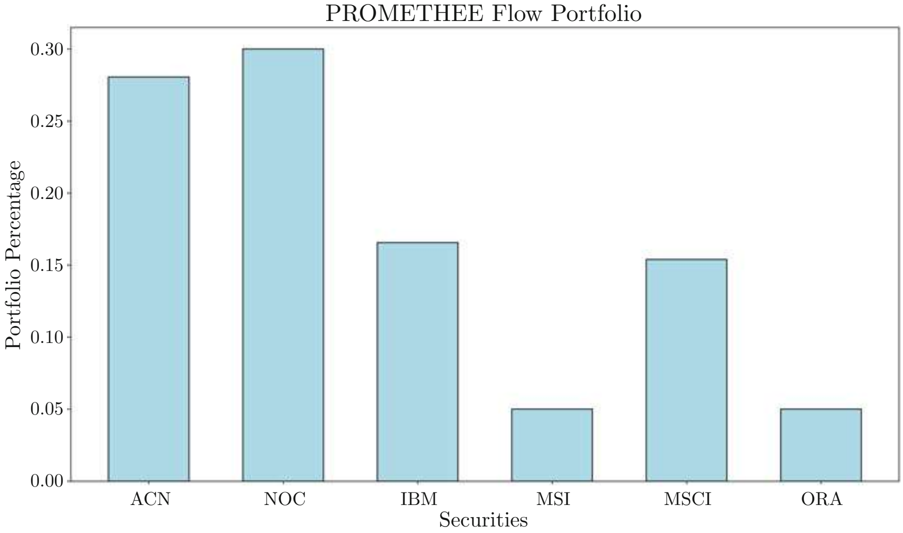
```
OptimizationStatus.OPTIMAL

Maximum DY =  2.5395
```

Finally, we construct the final problem as a goal programming optimization problem including the minimax objective Q. The results for a random selection of offsets are presented below:

[6]:
```
m = Model()

w1 = 0.2
w2 = 0.1
w3 = 0.7

onoff = [ m.add_var(var_type=BINARY) for i in␣
↪range(numSecurities) ]
weights = [ m.add_var(var_type=CONTINUOUS) for i in␣
↪range(numSecurities) ]
Q = m.add_var(var_type=CONTINUOUS)

m += xsum(weights[i] for i in range(numSecurities))␣
↪== 1
m += xsum(onoff[i] for i in range(numSecurities)) <=␣
↪maxSecurities
m += xsum(onoff[i] for i in range(numSecurities)) >=␣
↪minSecurities
for i in range(numSecurities):
    m += weights[i] - lowerBound * onoff[i] >= 0
    m += weights[i] - upperBound * onoff[i] <= 0

m += w1 * ((xsum(weights[i] * betaIndex[i] for i in␣
↪range(numSecurities)) - minBeta) / minBeta <= Q
m += w2 * (maxProm - (xsum(weights[i] * promIndex[i]␣
↪for i in range(numSecurities)))) / maxProm <= Q
```

```
m += w3 * (maxDY - (xsum(weights[i] * DYIndex[i] for␣
↪i in range(numSecurities)))) / maxDY <= Q


m.objective = minimize(Q)

status = m.optimize()

plt.figure(figsize=(16,9))
for i in range(numSecurities):
    plt.bar(companyName[i], weights[i].x, color =␣
↪'lightblue', edgecolor = 'black', width=0.6)
plt.xlabel(r"Securities", fontsize=16)
plt.ylabel(r"Portfolio Percentage", fontsize=16)
plt.title(r"PROMETHEE Flow Portfolio", fontsize=21)
plt.xticks(fontsize=12, rotation=0)
```

```
OptimizationStatus.OPTIMAL

Q =  0.03350039947771772

ACN :  0.2805866288365215
NOC :  0.3
IBM :  0.16550896312890123
MSI :  0.04999999999999996
MSCI :  0.15390440803457714
ORA :  0.05

Portfolio beta =  1.005219219751575
Portfolio PROMETHEE flow =  0.056381686302817036
Portfolio DY =  2.417965336466194
```



6.6 Conclusions

The information system presented in this chapter aims to include the entire decision-making process. The most important part, however, is that any subsystem within the general information system can operate independently. Thus, the user is not limited to follow a specific route, but he can use the system tailored to his own special needs. For instance, he may select his own specific criteria for the MCDA portfolio selection, he may decide to utilize only one specific method for the multiobjective portfolio optimization, or even decide to use one of the two phases independently. In this sense, we facilitate the need to incorporate the personal preferential system of the decision-maker into the decision-making process.

Chapter 7
Empirical Testing

7.1 Introduction

In this chapter we present the empirical testing of the proposed methodology on real data. This step is very important, because it is necessary that the methodology is explicitly tested to verify its validity. Consequently, a large experimental application of the proposed methodological framework was conducted, including securities from four stock exchanges (NYSE, NASDAQ, Paris, and Tokyo). Thus, every step of the methodological framework is examined, while the results are tested with an out-of-sample validation process.

The input data for the first phase of the process (incl. financial indexes of the securities) were drawn from www.investing.com. The input data for the second phase of the process (incl. historical data for time horizon of 3 years) were drawn from the www.finance.yahoo.com.

In the second section of the chapter, we present the main characteristics of the application field. The third section of the chapter includes a detailed description of the empirical testing process for NYSE stock exchange. This section is divided into two parts. The first part describes the first phase of the methodology, while the second part presents the results of the second phase of the methodology. Finally, the fourth section contains the out-of-sample validation procedure of the results.

It is important to emphasize that an extensive presentation of the input data and a large part of the results are presented in Appendix A.

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_7

7.2 Empirical Testing Information

The proposed methodology was applied to four stock exchanges: (a) *NYSE*, (b) *NASDAQ*, (c) *Paris*, and (d) *Tokyo*. The total number of the examined securities is about two thousand, while the time horizon of the analysis was set to four calendar years.

The set of the examined securities was split according to the industrial sector and the stock exchange of each security. The companies which participated in the empirical testing process belong to three industrial sectors: (a) *technological*, (b) *energy*, and (c) *financial*. The input data for the first phase were fetched from the www.investing.com database. However, for a large number of securities there were insufficient data. Therefore, the companies that did not satisfy the requirements (missing data, zero values, etc.) were excluded from the experiment. In Table 7.1 we record the securities of each stock exchange, split according to their industrial sector. In the last column we present the total number of securities in each sector (including the securities with insufficient data).

7.3 Results Presentation

In this section there is a detailed description of the input data and the obtained results of the empirical testing process during each phase of the methodological framework. It is necessary to mark the fact that this section refers to the NYSE stock exchange experimental results.

Table 7.1 Empirical testing input information

Stock exchange	Industrial sector	Number of experiment securities	Number of securities with missing data	Total number of securities
NYSE	technology	69	177	246
	energy	89	131	220
	financial	358	461	819
NASDAQ	technology	326	213	539
	energy	6	40	46
	financial	93	471	564
Paris	technology	50	91	141
	energy	7	8	15
	financial	33	24	57
Tokyo	technology	485	263	748
	energy	30	4	34
	financial	143	51	194

7.3.1 Phase I: Multicriteria portfolio selection

The first phase of the empirical testing includes the portfolio selection process, based on multicriteria decision analysis methods. The aim of this phase is to locate the securities which have a strong evolutionary potential. The process is based on four multicriteria decision analysis methods.

The input values that must be determined in order to perform the four ranking methods include the *evaluation matrix* which contains the performance of each alternative in the determined financial criteria, as well as the *offsets* and the *thresholds* for each criterion.

The selection of the offsets was determined according to three different scenarios, in order to conduct a sensitivity analysis on the results. In this paragraph, the results according to the scenario of equal offsets among the alternatives are presented.

The thresholds configuration differs significantly according to the multicriteria method. For each ranking method the configuration process was based on the partition of the values' range. Firstly, we determine the range by calculating the minimum and maximum value of the alternatives for each criterion and secondly we split this range as follows: (i) For ELECTRE III method which involves three different thresholds (preference p, indifference q, and veto v) we split the range into four sections and assign the values, respectively: $q(i) < p(i) < v(i)$, (ii) For PROMETHEE II method which involves two different thresholds (preference p and indifference q) we split the range into three sections and assign the values, respectively: $q(i) < p(i)$, (iii) MAUT and TOPSIS methods do not involve any thresholds.

Given the results of the multicriteria methods, the final step of the first phase involves the cumulative ranking of the securities. Assuming that we select the 20-highest ranked securities from each industrial sector, the portfolio has been formulated as shown in Table 7.2.

7.3.2 Phase II: Multiobjective Portfolio Optimization

The second phase of the methodological framework includes the portfolio optimization process. Therefore, the objective of this phase is the determination of the proportion of each security in the portfolio, given a set of securities which were selected in phase I. The results of the first phase (identification of the securities which serve as the best investment prospects) are the input for the second phase of the proposed methodology.

The portfolio includes securities from all three industrial sectors. Therefore, the number of securities that will participate in the second phase of the process is 60. More specifically, we select the 20 most favorable companies from each industrial sector. Phase II includes four different methods for portfolio optimization: (a) mean–variance MIQP model, (b) goal programming model, (c) PROMETHEE flow multiobjective model, and (d) genetic algorithm model.

Table 7.2 Selected securities from NYSE stock exchange

Technology	Energy	Financial
Northrop Grumman	Phillips 66	RenaissanceRe
GlobalSCAPE	NACCO Industries	White Mountains Insurance
Accenture	Cypress Energy Partners LP	Triplepoint Venture
Synnex	Global Partners	JPMorgan
IBM	Sunoco LP	Cohen Steers TR Realty Closed
Taiwan Semiconductor	TC Energy	Santander Consumer USA Holdings Inc
Motorola	Royal Dutch Shell ADR	BlackRock Taxable Muni Bond Trust
Jabil Circuit	GasLog Partners Pref A	Hartford
Oracle	Phillips 66 Partners LP	Wells Fargo Real Estate Invest Pref
MSCI	Royal Dutch Shell B ADR	Nuveen AMT Free Muni Credit
Roper Technologies	World Fuel Services	MFS California
Danaher	Cosan Ltd	Wells Fargo Pref L
Leidos	Magellan	Saratoga Investment Corp
Benchmark Electronics	CVR Energy	Allstate
Infosys ADR	BP ADR	Blackrock Muni Target Term Closed
Hubbell	Chevron	PennyMac Mortgage
Nelnet	CNOOC ADR	Flaherty and Crumrine Dynamic Pref
CAE Inc.	Exxon Mobil	Metlife Inc Pref
Hexcel	ONEOK	PNC Financial
Broadridge	PetroChina ADR	Reinsurance of America

Method 1: Mean–Variance MIQP Model

The first method is based on a variation of the mean–variance approach, which is extended with additional constraints. More specifically, the imposed constraints are the following:

1. Minimum number of securities to participate in a portfolio equal to 4.
2. Maximum number of securities to participate in a portfolio equal to 40 of the total number of securities.
3. Minimum percentage of capital invested in a security (if this security participates to the portfolio) equal to 0.05%.
4. Maximum percentage of capital invested in a security (if this security participates to the portfolio) equal to 25%.
5. Minimum percentage of capital invested on a specific industrial sector equal to 5%.
6. Maximum percentage of capital invested on a specific industrial sector equal to 50%.

Table 7.3 shows the composition of each Pareto optimal portfolio, i.e. the percentage of capital invested in each security.

Table 7.3 Set of efficient portfolios for NYSE stock exchange

Portf	NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI	ROP	…	DFP	MET	PNC	RGA
1	0.01	0.00	0.0	0.0	0.0	0.0	0.01	0.0	0.0	0.00	0.0	…	0.08	0.0	0.00	0.00
2	0.01	0.00	0.0	0.0	0.0	0.0	0.01	0.0	0.0	0.00	0.0	…	0.08	0.0	0.01	0.00
3	0.01	0.01	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.00	0.0	…	0.07	0.0	0.01	0.00
4	0.00	0.01	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.00	0.0	…	0.06	0.0	0.00	0.00
5	0.01	0.01	0.0	0.0	0.0	0.0	0.03	0.0	0.0	0.00	0.0	…	0.06	0.0	0.00	0.00
6	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	…	0.05	0.0	0.00	0.01
7	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	…	0.04	0.0	0.00	0.02
8	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	…	0.03	0.0	0.00	0.02
9	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	…	0.00	0.0	0.00	0.03
10	0.00	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.00	0.0	…	0.00	0.0	0.00	0.03
11	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.01	0.0	…	0.00	0.0	0.00	0.03
12	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.01	0.0	…	0.00	0.0	0.00	0.03
13	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.02	0.0	…	0.00	0.0	0.00	0.03
14	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.02	0.0	…	0.00	0.0	0.00	0.03
15	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.03	0.0	…	0.00	0.0	0.00	0.03
16	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.03	0.0	…	0.00	0.0	0.00	0.04
17	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.04	0.0	…	0.00	0.0	0.00	0.04
18	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.05	0.0	…	0.00	0.0	0.00	0.04
19	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.05	0.0	…	0.00	0.0	0.00	0.04
20	0.00	0.01	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.06	0.0	…	0.00	0.0	0.00	0.05
21	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.06	0.0	…	0.00	0.0	0.00	0.05
22	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.06	0.0	…	0.00	0.0	0.00	0.06
23	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.07	0.0	…	0.00	0.0	0.00	0.06
24	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.07	0.0	…	0.00	0.0	0.00	0.07
25	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.08	0.0	…	0.00	0.0	0.00	0.06

(continued)

Table 7.3 (continued)

Portf	NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI	ROP	...	DFP	MET	PNC	RGA
26	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.08	0.0	...	0.00	0.0	0.00	0.06
27	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.09	0.0	...	0.00	0.0	0.00	0.07
28	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.10	0.0	...	0.00	0.0	0.00	0.07
29	0.00	0.01	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.11	0.0	...	0.00	0.0	0.00	0.07
30	0.00	0.01	0.0	0.0	0.0	0.0	0.07	0.0	0.0	0.12	0.0	...	0.00	0.0	0.00	0.07
31	0.00	0.00	0.0	0.0	0.0	0.0	0.07	0.0	0.0	0.13	0.0	...	0.00	0.0	0.00	0.07
32	0.00	0.00	0.0	0.0	0.0	0.0	0.07	0.0	0.0	0.14	0.0	...	0.00	0.0	0.00	0.07
33	0.00	0.00	0.0	0.0	0.0	0.0	0.07	0.0	0.0	0.15	0.0	...	0.00	0.0	0.00	0.07
34	0.00	0.00	0.0	0.0	0.0	0.0	0.07	0.0	0.0	0.16	0.0	...	0.00	0.0	0.00	0.07
35	0.00	0.00	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.17	0.0	...	0.00	0.0	0.00	0.06
36	0.00	0.00	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.18	0.0	...	0.00	0.0	0.00	0.06
37	0.00	0.00	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.19	0.0	...	0.00	0.0	0.00	0.05
38	0.00	0.00	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.21	0.0	...	0.00	0.0	0.00	0.03
39	0.00	0.00	0.0	0.0	0.0	0.0	0.01	0.0	0.0	0.22	0.0	...	0.00	0.0	0.00	0.02
40	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.24	0.0	...	0.00	0.0	0.00	0.01
41	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.23	0.0	...	0.00	0.0	0.00	0.00
42	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.23	0.0	...	0.00	0.0	0.00	0.00
43	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.22	0.0	...	0.00	0.0	0.00	0.00
44	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.21	0.0	...	0.00	0.0	0.00	0.00
45	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.21	0.0	...	0.00	0.0	0.00	0.00
46	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.23	0.0	...	0.00	0.0	0.00	0.00
47	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.19	0.0	...	0.00	0.0	0.00	0.00
48	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.15	0.0	...	0.00	0.0	0.00	0.00
49	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.11	0.0	...	0.00	0.0	0.00	0.00
50	0.00	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.08	0.0	...	0.00	0.0	0.00	0.00

Table 7.4 Goal programming portfolio for NYSE stock exchange

NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI
0.1158	0.0300	0.0	0.0	0.0	0.0	0.0300	0.0	0.2000	0.0
ROP	DHR	LDOS	BHE	INFY	HUBB	NNI	CAE	HXL	BR
0.0300	0.0299	0.0	0.0443	0.0356	0.0	0.0300	0.0300	0.0	0.0305
PSX	NC	CELP	GLP	SUN	TRP	RDS-A	GLOP	PSXP	RDS-B
0.0	0.0300	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
INT	CZZ	MMP	CVI	BP	CVX	CEO	XOM	OKE	PTR
0.0300	0.0300	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RNR	WTM	TPVG	JPM	RFI	SC	BBN	HIG	NVG	WFC
0.0300	0.0300	0.0	0.0	0.0	0.0	0.0	0.0300	0.0	0.0
SAR	ALL	BTT	PMT	DFP	MET	PNC	RGA		
0.0	0.1541	0.0300	0.0	0.0	0.0	0.0	0.0300		

Method 2: Goal Programming Model

The second method is a goal programming model, with the following constraints:

1. Goal 1: Portfolio beta set equal to 0.9
2. Goal 2: Portfolio dividend yield set equal to 1.5%
3. Goal 3: Percentage of securities with revenue $\geq$ 30 billions set equal to 50%.
4. Minimum percentage of capital invested in a security (if this security participates to the portfolio) equal to 0.03%.
5. Maximum percentage of capital invested in a security (if this security participates to the portfolio) equal to 20 %.
6. Minimum number of securities to participate in a portfolio equal to 20.
7. Maximum number of securities to participate in a portfolio equal to 40.

The three goals were equipped with deviational variables, while the four constraints are strict. The goal programming model resulted in the portfolio presented in Table 7.4.

Method 3: PROMETHEE Flow Multiobjective Programming Model

The third method is a bi-objective programming model which includes two objective functions: (a) the PROMETHEE net flow of the alternatives and (b) the portfolio beta. The model is equipped with the following constraints:

1. Minimum percentage of capital invested in a security (if this security participates to the portfolio) equal to 0.03%.
2. Maximum percentage of capital invested in a security (if this security participates to the portfolio) equal to 20%.
3. Minimum number of securities to participate in a portfolio equal to 20.
4. Maximum number of securities to participate in a portfolio equal to 40.

The problem was solved parametrically, setting portfolio beta as a parameter. The efficient frontier is presented in the following figure.

The efficient portfolios are presented in the following table.

Method 4: Genetic Algorithm Model

Finally, the fourth method is a genetic algorithm model. This method differs from the previous methods, as it is a passive strategy for portfolio optimization. More specifically, the returns of the securities are compared to the returns of the market index. The target is to maximize the number of time periods that the constructed portfolio beats the market index.

The resulting portfolio of this method is presented in the following table.

7.4 Out-of-Sample Validation

Generally, the comparison of the output of the proposed methodological framework to the market returns is a necessary procedure in order to evaluate the accuracy of the methodology. This procedure is vital for the confirmation of the produced output. It is obvious that the comparison must take place after the moment of the investment. Therefore, the goal of the validation is to prove that the produced portfolios perform equally good or better compared to the market indexes, for a period after the analysis time horizon.

Therefore, the validation process is based on out-of-sample data, which do not belong to the initial set of input data which were used during the analysis. Thus, the portfolio optimization procedure was based on daily data for the time period from 01/01/2016 until 31/12/2018. The validation process takes place in the following time period, from 01/01/2019 until 30/06/2019. The comparison was conducted for three different time periods: (a) short-term (1 month), (b) mid-term (3 months), (c) long-term (6 months).

7.4.1 NYSE Stock Exchange

The results of the validation process for NYSE stock exchange are presented in Tables 7.5, 7.6 and 7.7. Additionally, Figure 7.1 includes a comparative graphical representation of the expected daily capital return (%) for the selected optimal portfolio with every optimization model compared to the market index. The empirical testing procedure for NYSE stock exchange resulted in the following findings:

Table 7.5 Set of efficient portfolios for NYSE stock exchange with MOIP PROMETHEE method

Portf	NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI	ROP	…	DFP	MET	PNC	RGA
1	0.00	0.03	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	…	0.03	0.00	0.00	0.03
2	0.03	0.03	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
3	0.03	0.03	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
4	0.03	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
5	0.09	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
6	0.14	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
7	0.18	0.03	0.03	0.03	0.00	0.03	0.03	0.00	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
8	0.20	0.03	0.03	0.03	0.00	0.03	0.03	0.03	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
9	0.20	0.03	0.03	0.03	0.00	0.03	0.03	0.03	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
10	0.20	0.03	0.03	0.08	0.00	0.03	0.03	0.03	0.00	0.00	0.00	…	0.03	0.00	0.00	0.00
11	0.20	0.03	0.03	0.10	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
12	0.20	0.03	0.03	0.11	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
13	0.20	0.03	0.03	0.16	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
14	0.20	0.03	0.03	0.17	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
15	0.20	0.03	0.03	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
16	0.20	0.03	0.04	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
17	0.20	0.03	0.04	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
18	0.20	0.03	0.09	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
19	0.20	0.03	0.09	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00

(continued)

Table 7.5 (continued)

Portf	NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI	ROP	…	DFP	MET	PNC	RGA
20	0.20	0.00	0.08	0.20	0.00	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
21	0.20	0.00	0.04	0.20	0.03	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
22	0.20	0.00	0.04	0.20	0.03	0.03	0.03	0.03	0.00	0.03	0.00	…	0.00	0.00	0.00	0.00
23	0.18	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	…	0.00	0.00	0.00	0.00
24	0.13	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	…	0.00	0.00	0.00	0.00
25	0.09	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	…	0.00	0.00	0.00	0.00
26	0.04	0.00	0.03	0.20	0.03	0.03	0.00	0.03	0.03	0.03	0.00	…	0.00	0.00	0.00	0.00
27	0.03	0.00	0.03	0.17	0.03	0.03	0.00	0.03	0.03	0.03	0.03	…	0.00	0.00	0.00	0.00
28	0.03	0.00	0.03	0.10	0.03	0.03	0.00	0.03	0.03	0.03	0.03	…	0.00	0.00	0.00	0.00
29	0.03	0.00	0.03	0.04	0.03	0.03	0.00	0.03	0.03	0.03	0.03	…	0.00	0.00	0.00	0.00
30	0.00	0.00	0.03	0.00	0.03	0.00	0.00	0.00	0.03	0.03	0.03	…	0.00	0.03	0.03	0.00

Table 7.6 Genetic algorithm portfolio for NYSE stock exchange

NOC	GSB	ACN	SNX	IBM	TSM	MSI	JBL	ORCL	MSCI
0.1	0.005	0.003	0.209	0.004	0.0	0.003	0.005	0.003	0.004
ROP	DHR	LDOS	BHE	INFY	HUBB	NNI	CAE	HXL	BR
0.322	0.003	0.003	0.005	0.004	0.004	0.003	0.004	0.196	0.003
PSX	NC	CELP	GLP	SUN	TRP	RDS-A	GLOP	PSXP	RDS-B
0.003	0.0	0.0	0.0	0.003	0.004	0.003	0.005	0.005	0.004
INT	CZZ	MMP	CVI	BP	CVX	CEO	XOM	OKE	PTR
0.003	0.004	0.002	0.003	0.003	0.004	0.004	0.003	0.002	0.004
RNR	WTM	TPVG	JPM	RFI	SC	BBN	HIG	NVG	WFC
0.003	0.003	0.004	0.004	0.004	0.004	0.003	0.003	0.005	0.004
SAR	ALL	BTT	PMT	DFP	MET	PNC	RGA		
0.004	0.002	0.004	0.003	0.003	0	0	0.003		

Table 7.7 Expected daily capital return (%) for selected optimal portfolios and market index (NYSE stock exchange)

Time Horizon	Market Index	MIQP Mean–Variance	Goal Programming	MOIP PROMETHEE flow	Genetic Algorithm
1 month	0.4048	0.441	0.4874	0.6948	0.468
3 months	0.2049	0.2648	0.2836	0.2932	0.3135
6 months	0.1182	0.2565	0.2336	0.2353	0.2119

- During the period of January 2019 (1-month time horizon), all four models perform better than NYSE market index. The PROMETHEE flow model seems to offer the best expected return compared to the other models, while the MIQP mean–variance model offers the lowest expected return.
- During the period of January–March 2019 (3-month time horizon), the genetic algorithm model seems to perform better than the other models. However, it is important to note that all the models offer a higher expected return compared to the market index.
- Finally, during the period of January–June 2019 (6-month time horizon), all four models offer similar expected return, which is significantly higher than the market index return. Among the four models, the MIQP mean–variance model seems to offer slightly better results for a long term.

7.4.2 NASDAQ Stock Exchange

The results of the validation process for NASDAQ stock exchange are presented in Table 7.8. Additionally, Figure 7.2 includes a comparative graphical representation of the expected daily capital return (%) for the selected optimal portfolio with every

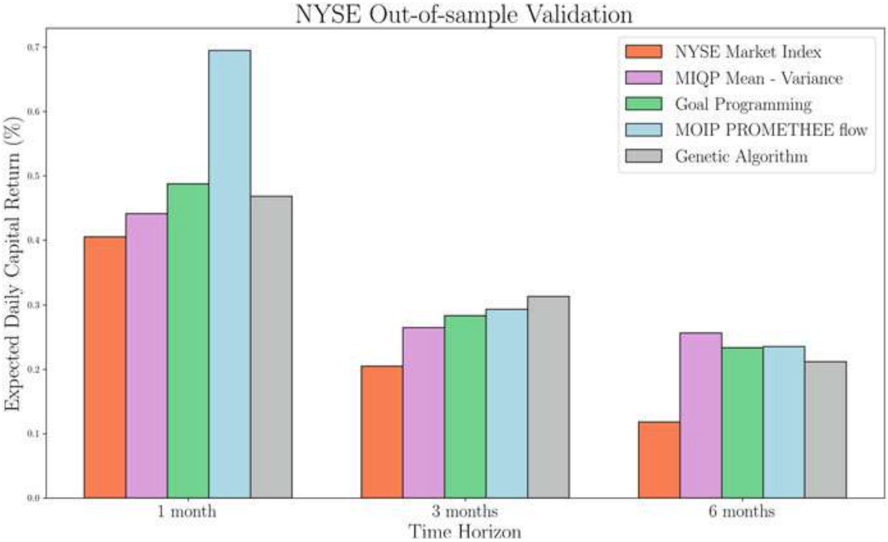


Fig. 7.1 Comparative graphical representation of expected daily capital return (%) for selected optimal portfolio and market index (NYSE stock exchange)

Table 7.8 Expected daily capital return (%) for selected optimal portfolios and market index (NASDAQ stock exchange)

Time Horizon	Market Index	MIQP Mean–Variance	Goal Programming	MOIP PROMETHEE flow	Genetic Algorithm
1 month	0.4612	0.3318	0.2924	0.3719	0.3456
3 months	0.2914	0.1996	0.2938	0.2861	0.2807
6 months	0.1712	0.1904	0.2081	0.2766	0.1847

optimization model compared to the market index. The empirical testing procedure for NASDAQ stock exchange resulted in the following findings:

- During the period of January 2019 (1-month time horizon), we observe that all four models perform better than NASDAQ market index. The goal programming model seems to offer the best expected return compared to the other models, while the MIQP mean–variance model offers the lowest expected return.
- During the period of January–March 2019 (3-month time horizon), we note that all four models offer significantly better return than NASDAQ market index. Among the four models, the genetic algorithm model seems to perform worse than the other models, while the other three models offer similar expected return.
- Finally, during the period of January–June 2019 (6-month time horizon), it is obvious that the goal programming model offers the best result, while the PROMETHEE flow model performs equally well. As in the previous cases, all four models are more profitable than the market index.

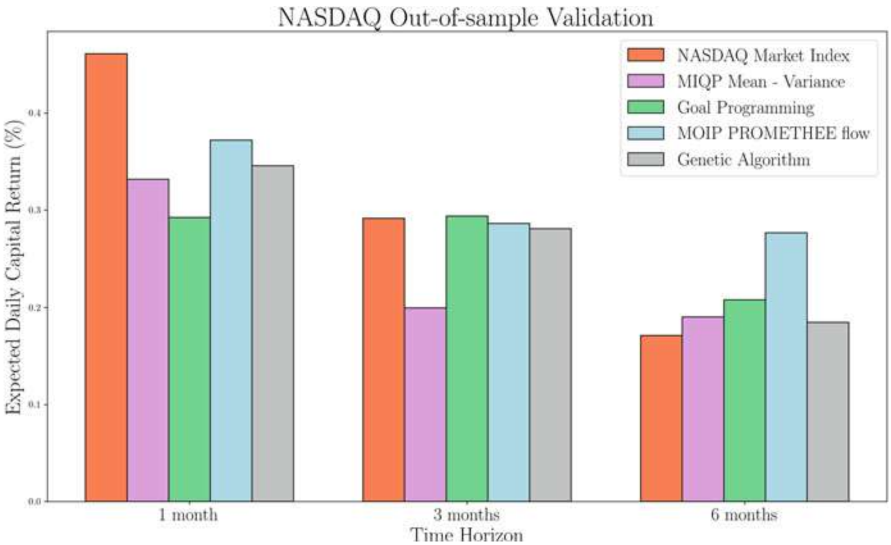


Fig. 7.2 Comparative graphical representation of expected daily capital return (%) for selected optimal portfolio and market index (NASDAQ stock exchange)

7.4.3 PARIS Stock Exchange

The results of the validation process for Paris stock exchange are presented in Table 7.9. Additionally, Figure 7.3 includes a comparative graphical representation of the expected daily capital return (%) for the selected optimal portfolio with every optimization model compared to the market index. The empirical testing procedure for Paris stock exchange resulted in the following findings:

- During the period of January 2019 (1-month time horizon), only the goal programming model performs better than CAC40 market index, while the genetic programming model performs equally well. On the other hand the PROMETHEE flow and the mean–variance models seem to offer lower expected return compared to the market.
- During the period of January–March 2019 (3-month time horizon), the goal programming model seems to perform better than all the other models, while the genetic algorithm model offers similar expected return compared to the market index.
- Finally, during the period of January–June 2019 (6-month time horizon), the genetic algorithm model offers the greatest expected return. The only model that performs slightly worse than the market index is the MIQP mean–variance model.

Table 7.9 Expected daily capital return (%) for selected optimal portfolios and market index (PARIS stock exchange)

Time Horizon	Market Index	MIQP Mean–Variance	Goal Programming	MOIP PROMETHEE flow	Genetic Algorithm
1 month	0.3517	0.2327	0.3864	0.2357	0.3384
3 months	0.2285	0.1852	0.3118	0.2355	0.1721
6 months	0.1513	0.0938	0.2391	0.1972	0.2692

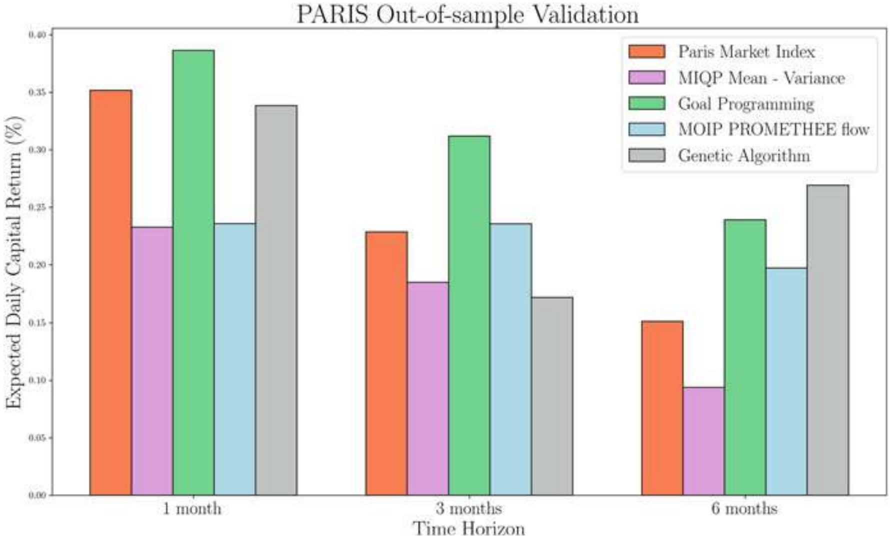


Fig. 7.3 Comparative graphical representation of expected daily capital return (%) for selected optimal portfolio and market index (PARIS stock exchange)

7.4.4 TOKYO Stock Exchange

The results of the validation process for Tokyo stock exchange are presented in Table 7.10. Additionally, Figure 7.4 includes a comparative graphical representation of the expected daily capital return (%) for the selected optimal portfolio with every optimization model compared to the market index. The empirical testing procedure for Tokyo stock exchange resulted in the following findings:

- During the period of January 2019 (1-month time horizon), all four models perform better than Nikkei225 market index. The goal programming model seems to offer the best expected return compared to the other models, while the MIQP mean–variance and the genetic algorithm model offer the lowest expected return.

Table 7.10 Expected daily capital return (%) for selected optimal portfolios and market index (TOKYO stock exchange)

Time Horizon	Market Index	MIQP Mean–Variance	Goal Programming	MOIP PROMETHEE flow	Genetic Algorithm
1 month	0.3277	0.5681	0.8468	0.6715	0.5368
3 months	0.139	0.3829	0.4103	0.3963	0.2942
6 months	0.0829	0.2032	0.288	0.2616	0.1368

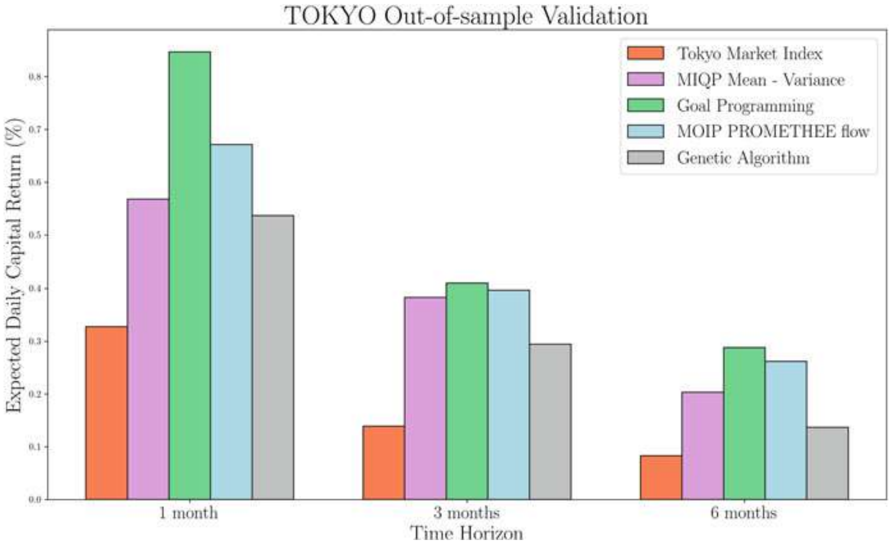


Fig. 7.4 Comparative graphical representation of expected daily capital return (%) for selected optimal portfolio and market index (TOKYO stock exchange)

- During the period of January–March 2019 (3-month time horizon), all four models continue to perform better than Nikkei225 market index. More specifically, the goal programming model seems to perform slightly better than the other models, while the genetic algorithm model offers the lowest expected return.
- Finally, during the period of January–June 2019 (6-month time horizon), all four models offer higher expected return compared to the market index. Among the four models, the goal programming model seems to offer slightly better results for a long term.

7.5 Conclusions

Conclusively, the experimental application on real data certifies the validity of the proposed methodological framework. During the process of the experimental application we should introduce a series of logical decisions (number of securities participating in the portfolio, desired percentage of companies from each industry, etc.) which refer to a neutral investor's profile. As shown in the out-of-sample presentation section, the portfolios that were generated by the proposed methodology perform equally good or even better than the market index, thus rendering the proposed methodology a reliable tool in the hands of an investor.

Chapter 8
Concluding Remarks and Future Prospects

In conclusion, we highlighted the inability of conventional methods to deal effectively with the portfolio management problem due to many reasons, such as the failure of combining the multiple conflicting criteria which must be considered during the portfolio construction process, the inadequacy to integrate the preferential profile of the decision-maker during the decision-making process, and the lack of flexibility to allow the user to intervene in the process and set his own policy restrictions.

Having acknowledged that the portfolio management process is a problem that combines many criteria, we presented a modern, integrated methodological framework based on the scientific field of multicriteria decision analysis. Therefore, we divided the problem into two distinct phases (security selection and portfolio optimization) and we applied a series of multiobjective algorithms to each step of the process. More specifically, the first phase involves comparing a set of securities with four multicriteria ranking methods and results in the predominant securities based on the criteria introduced by the user. The second phase involves finding the most effective portfolios consisting of these securities. The most unique feature of the proposed methodology is that it manages to incorporate a series of criteria and policy restrictions. Additionally, the main contribution of our project was the development of an integrated information system, where we implemented all the algorithms of the proposed methodology in the Python programming language.

The contribution of the proposed methodology in the portfolio construction process is significant, as it supports financial analysts, portfolio managers, and investors in comparing sets of securities and managing their portfolios. However, the financial markets are rapidly changing bringing new opportunities and challenges in the fields of multicriteria analysis and portfolio management. Moreover, new technologies have emerged which have resulted in the improvement of the existing information systems and the emergence of opportunities for interoperable systems. Regarding all the above, a series of future prospects can be considered, such as:

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2_8

- Connecting the information system with other smart systems, in order to automate the portfolio management process. The information system, as designed now, efficiently supports the decision-making process, but cannot automatically perform processes such as reporting and order execution. In this direction, it would be very important for the system to become interoperable with existing web-based applications. It is self-evident, of course, that such a perspective raises important issues, such as security issues, data synchronization issues, user-specific accounts, etc.
- Expanding the proposed methodological framework in order to include additional asset classes (bonds, commodities, etc.). A modern approach requires combining different categories of assets, with the aim of increasing portfolio diversification, but also detecting investment opportunities in different categories of securities. However, each asset class contains different criteria for evaluating alternatives. Therefore, in order for the proposed model to incorporate all different asset classes, it is necessary to take into account the different criteria per class.
- Improving the computational complexity of the existing models, so that they can handle large volumes of data in real time. This could be achieved either by improving the existing algorithm of the utilized methods, or by distributing the calculations into multiple sources through distributed systems, which deal with big data.

Appendix A
Extensive Experimental Results

The objective of this appendix is to provide a series of tables which supplement the experimental results presentation and describe the application of the methodology step-by-step.

In Tables A.1–A.4, there is a presentation of the securities of each stock exchange. More specifically, the securities of NYSE stock exchange are recorded in Table A.1, the securities of NASDAQ stock exchange are recorded in Table A.2, the securities of Paris stock exchange are recorded in Table A.3 and, finally, the securities of Tokyo stock exchange are recorded in Table A.4. These tables include only the securities which were used in the empirical testing process (companies with missing data are not recorded). The first column includes the securities of the technological sector, the second column includes the securities of the energy sector, and the third column includes the securities of the financial sector.

In Tables A.5–A.7 the input data for each industrial sector are presented. More specifically, the companies of the energy sector are recorded in Table A.5, the companies of the financial sector are recorded in Table A.6, and finally the companies from the technological sector are recorded in Table A.7.

In Tables A.8–A.10, the results of each multicriteria method are presented for each sector. More specifically, the results for the energy sector are presented in A.8, the results for the technological sector are presented in A.9, and the results for the financial sector are presented in A.10.

Table A.11 includes the most significant financial indexes for the selected securities, including the minimum, maximum, and median value of the securities, the mean value, and the standard deviation, as well as the value-at-risk, the skewness, and the kurtosis of the securities:

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2

Table A.1 List of NYSE securities used in empirical testing

	Technology	Energy	Financial
1	ABB ADR	Petroleo Brasileiro ADR Reptg 2 Pref	Nuveen CA MVF 2
2	Accenture	Phillips 66	Nuveen High Income 2020 Target Term
3	SAP ADR	Phillips 66 Partners LP	Nuveen Dow 30Sm
4	Infosys ADR	Baker Hughes A	Ellsworth Growth Pref A
5	Wipro ADR	GasLog Partners Pref A	Federal Agricultural Mortgage A
6	BT ADR	Adams Resources & Energy	Chimera Investment Pref A
7	STMicroelectronics ADR	Ecopetrol ADR	Ares Management Pref A
8	Canon ADR	Total ADR	Apollo Global Management A
9	Agilent Technologies	Petroleo Brasileiro Petrobras ADR	Ladder Capital A
10	Allegion PLC	CNOOC ADR	Aberdeen Emerging Markets Equity
11	Ametek	Sinopec Shanghai Petrochemical ADR	Aberdeen Asia-Pacific
12	Amphenol	Royal Dutch Shell ADR	Adams Diversified Equity Closed
13	AO Smith	Equinor ADR	Barclays ADR
14	Scnc App In	ENI ADR	Santander Chile ADR
15	Rockwell Automation	PetroChina ADR	Sumitomo Mitsui Financial ADR
16	AVX	Transportadora Gas ADR	Mitsubishi UFJ Financial ADR
17	AZZ	BP ADR	China Life Insurance ADR
18	Badger Meter	Royal Dutch Shell B ADR	Aegon ADR
19	Belden	Plains All American Pipeline	Banco Bilbao ADR
20	Regal Beloit	YPF Sociedad Anonima	Credit Suisse ADR
21	Benchmark Electronics	Archrock	Prudential Public ADR
22	Broadridge	Teck Resources B	Lloyds Banking ADR
23	BWX Tech	BP Prudhoe Bay Royalty Trust	ING ADR
24	CAE Inc.	Cabot Oil & Gas	BBVA Banco Frances ADR
25	Jabil Circuit	Canadian Natural	Santander ADR
26	TE Connectivity	Cenovus Energy Inc	Itau CorpBanca ADR
27	Issuer Direct Corp	Chevron	Westpac Banking ADR
28	CTS Corp	Cimarex Energy	Nuveen California Div Advantag Muni
29	Danaher	CONSOL Coal	BlackRock Long Term Muni Advantage

(continued)

Table A.1 (continued)

	Technology	Energy	Financial
30	Deluxe	Concho Resources	Aflac
31	DXC Technology	ConocoPhillips	AG Mortgage Investment
32	Eaton	Continental Resources	AG Mortgage Invest Trust Pb Pref
33	Espey Mfg&Electronics	Cosan Ltd	AG Mortgage Invest Trust Pa Pref
34	Methode Electronics	Crestwood Equity Partners LP	Federal Agricultural Mortgage
35	Emerson	Crossamerica Partners LP	Great Ajax Corp
36	Energizer	CVR Energy	Alliance Data Systems
37	Enersys	Cypress Energy Partners LP	AllianceBernstein Holding LP
38	ESCO Technologies	Delek Logistics Partners LP	AllianzGI Diversifiedome Convertibl
39	Evertec Inc	Delek US Energy	AllianzGI Equity Convertible Closed
40	FactSet Research	Devon Energy	Ares Dynamic Credit Allocation Inc
...	...	...	...
69	Xerox	MPLX LP	Arthur J Gallagher
70		Suburban Propane Partners LP	Western Asset Mortgage
...		...	...
89		Williams	Great Western Bancorp Inc
90			Berkshire Hills Bancorp
...			...
358			Westwood

Table A.2 List of NASDAQ securities used in empirical testing

	Technology	Energy	Financial
1	Bel Fuse A	Diamondback	Nuveen NASDAQ 100 Dyn Over
2	Cognizant A	Alliance Resource	1st Source
3	Activision Blizzard	Viper Energy Ut	1st Constitution Bancorp
4	Formula Systems ADR	Dorchester Minerals	Bancorp 34
5	LM Ericsson B ADR	Hallador	National General A Pref
6	Allied Motion	TransGlobe Energy	Donegal A
7	Amdocs		ACNB
8	American Software		Hennessy Ad
9	Analog Devices		Grupo Financiero Galicia ADR
10	Apple		Alcentra Capital Corp
11	Applied Materials		Alerus Fin
12	Jack Henry&Associates		Amark Preci
13	AstroNova		America First Tax
14	AudioCodes		German American Bancorp
15	Hollysys Automation Tech		American National Bankshares
16	Avnet		American River
17	Bel Fuse B		Atlantic American
18	Blackbaud		American National Insurance
19	Broadcom		Ameris
20	Bruker		AMERISAFE
21	Cabot		AmeriServ
22	Camtek		TD Ameritrade
23	CDK Global Holdings LLC		Ames
24	CDW Corp		Apollo Invest
25	Cerner		Ares Capital
...	...		...
93	Xperi		Northwest Bancshares
94			Huntington Bancshares
...			...
329			Zions

Table A.3 List of Paris Stock Exchange securities used in empirical testing

	Technology	Energy	Financial
1	Akka	Total	BNP Paribas
2	Alten	TechnipFMC	AXA
3	Artois Nom.	Rubis	Credit Agricole
4	Atos	GTT	Societe Generale
5	Aubay	Total Gabon	Amundi
6	Aures Tech	Maurel et Prom	Natixis
7	Axway	Docks des Petroles dAmbes	CNP Assurances
8	Capgemini		SCOR
9	Cofidur		Euronext
10	Coheris		Eurazeo
11	CS Communication		FFP
12	Dassault Systemes		Rothschild & Co
13	Delfingen		CRCAM Langued
14	Devoteam		CRCAM Brie Picardie 2
15	DNXcorp		Coface
16	Schneider Electric		CRCAM Atlantique
17	Environnement		April
18	Esker		Crcam Touraine
19	Evolis		Crcam Ille-Vil
20	Fiducial Office		Altamir
21	GEA		Ca Toulouse 31 CCI
22	Perrier Gerard		Crcam Morbihan
23	ITS Group		Galimmo
24	Groupe Open		ABC Arbitrage
25	Guillemot		Viel Et Compagnie
26	Harvest		Union Financiere
27	Hitechpros		IDI
28	Infotel		Crcam Norm.Sei
29	Ingenico		Crcam Sud RA
30	Innelec		Lebon
31	Pharmagest Interactive		Crcam Loire Ht
32	Lacroix		Groupe IRD
33	Legrand		Idsud
34	Linedata Services		
35	Mersen		
36	Neurones		
37	Prodware		
38	Quadient		
39	Rexel		

Table A.4 List of Tokyo stock exchange securities used in empirical testing

	Technology	Energy	Financial
1	Yaskawa Electric Corp.	San-Ai Oil	The 77 Bank Ltd
2	Advantest Corp.	BP Castrol KK	Nihon M&A Center
3	Rohm Ltd	Mitsui Matsushima Co Ltd	Acom Co Ltd
4	Hitachi High-Technologies Corp	Idemitsu Kosan Co Ltd	Activia Properties
5	Nitto Denko Co	Sinanen Co Ltd	MS&AD Insurance Group Holdings
6	Shimadzu Corp	Itochu Enex Co Ltd	Advance Create
7	Otsuka Corp	Toell Co Ltd	Japan Investment Adviser
8	Disco Corp	Nippon Coke & Engineering Ltd	Aeon Financial Service Co Ltd
9	Trend Micro Inc.	Inpex Corp.	Aichi Bank Ltd
10	Ricoh	Marubeni Corp.	Aizawa Securities
11	Konami Corp.	Sojitz Corp.	Akatsuki
12	Itochu Techno Solutions	Sala Corp	Akita Bank Ltd
13	Hamamatsu Photonics KK	Kamei Corp	Anicom Holdings Inc
14	It Holdings Corp	Iwatani Corp	Anshin Guarantor Service
15	SCSK Corp	Tokai Holdings Corp	Aomori Bank Ltd
16	Nikon Corp.	MORESCO Corp	Asax Co Ltd
17	Brother Industries Ltd	Cosmo Energy Holdings	Ashikaga Holdings
18	Seiko Epson Cor	Daimaru Enawin	Astmax
19	KakakuCom Inc	K&O Energy Group Inc	Awa Bank Ltd
20	NGK Insulators	JP Petroleum Exploration Ltd	Yamanashi Chuo Bank
21	Alps Electric	Yamashin-Filter	Shiga Bank Ltd
22	Hirose Electric Co Ltd	Mitsuuroko Group Holdings	Kiyo Bank Ltd
23	Yokogawa Electric Corp.	Sumiseki Holdings Inc	Bank of Kochi Ltd
24	Fuji Electric	JX Holdings, Inc.	Chukyo Bank Ltd
25	SUMCO Corp.	Taiheiyo Kouhatsuorporated	Taiko Bank Ltd
26	Azbil Corp	Mitsui	Kita Nippon Bank
27	Konica Minolta, Inc.	Nissin Shoji	Yamagata Bank Ltd
28	Nihon Unisys Ltd	Toa Oil	Oita Bank Ltd
29	Lasertec Corp	Shinko Plantech	Miyazaki Bank Ltd
30	Taiyo Yuden	Sanrin	Ehime Bank Ltd
31	Ns Solutions Corp		Fukushima Bank Ltd
32	Ibiden Co Ltd		The Bank Of Kyoto Ltd
33	Dainippon Screen Mfg.		The Chugoku Bank Ltd
34	Obic Business Consultants		The Iyo Bank Ltd
35	Capcom Co Ltd		The Hiroshima Bank Ltd

(continued)

Table A.4 (continued)

	Technology	Energy	Financial
36	Koei Tecmo Holdings		Chiba Kogyo Bank
37	Canon Marketing Japan Inc		Bank of Iwate Ltd
38	DeNA Co		Michinoku Bank Ltd
39	Anritsu Corp		San-in Godo Bank
...	...		...
142	FTGroup		United Urban
143	Marvelous Inc		
...	...		
482	Obic Co Ltd		

Table A.5 Evaluation Table input data for NYSE Energy Sector

Decision Matrix	P/E Ratio	EPS	Rev (B)	Beta	DY(%)	Mon	YTD (%)	1 Year
Petroleo Brasileiro ADR Reptg 2 Pref	12.23	1.15	101.69	1.47	1.61	3	11.56	−6.44
Phillips 66	9.21	11.57	109.68	1.07	3.38	5	23.74	−2.61
Phillips 66 Partners LP	13.26	4.16	1.1	0.86	6.19	5	31.09	7.52
Baker Hughes A	170.91	0.13	23.54	1.02	3.3	1	1.4	−29.34
GasLog Partners Pref A	10.31	1.91	0.376	0.98	11.17	5	16.76	0.59
Adams Resources&Energy	40.77	0.73	1.84	0.78	3.21	1	−22.76	−25.88
Ecopetrol ADR	10.41	1.64	21.64	1.6	8.14	1	7.43	−33.1
Total ADR	12.39	3.96	185.04	0.75	5.92	1	−4.48	−19.06
Petroleo Brasileiro Petrobras ADR	12.25	1.15	101.69	1.47	1.61	2	8.3	−8.21
CNOOC ADR	8.24	18.09	34.02	1.07	6.25	1	−2.22	−18.59
Sinopec Shanghai Petrochemical ADR	7.57	3.8	14.14	0.99	12.63	1	−33.32	−41.24
Royal Dutch Shell ADR	11.58	4.94	376.66	0.87	6.57	1	−1.73	−12.26
Equinor ADR	7.49	2.47	74.02	0.99	5.3	1	−12.71	−31.25
ENI ADR	14.31	2.1	128.73	0.78	6.16	1	−4.44	−16.71
PetroChina ADR	12.39	4.14	360.8	1.16	7.21	1	−16.62	−31.94
Transportadora Gas ADR	4.57	1.75	0.952	0.89	22.68	1	−44.67	−39.68

(continued)

Table A.5 (continued)

Decision Matrix	P/E Ratio	EPS	Rev (B)	Beta	DY(%)	Mon	YTD (%)	1 Year
BP ADR	14.13	2.63	294.14	0.77	6.62	1	−2	−16.59
Royal Dutch Shell B ADR	11.58	4.94	376.66	0.87	6.57	1	−3.64	−14.19
Plains All American Pipeline	4.87	3.94	34.21	1.02	7.51	1	−4.39	−21.38
YPF Sociedad Anonima	8.36	1.1	12.49	1.46	2.37	1	−29.05	−35.55
Archrock	22.73	0.41	0.94	2.85	6.18	1	25.37	−17.78
Teck Resources B	4.45	3.34	9.61	1.51	1.02	1	−26.37	−31.73
BP Prudhoe Bay Royalty Trust	2.07	4.14	0.089	−0.2	26.03	1	−60.44	−76.49
Cabot Oil&Gas	9.08	1.93	2.44	0.54	2.06	1	−21.66	−25.11
Canadian Natural	8.28	3	15.45	1.21	3.96	1	3.11	−14.33
Cenovus Energy Inc	44.31	0.18	15.9	1.02	1.43	1	16.22	−6.74
Chevron	14.68	7.71	152.89	1.01	4.21	2	5.33	−2.7
Cimarex Energy	7.24	6.19	2.34	1.36	1.78	1	−27.23	−51.78
CONSOL Coal	7.2	1.75	0.34	1.06	16.26	1	−23.71	−30.94
Concho Resources	26.24	2.45	4.49	1.26	0.78	1	−37.55	−57.33
ConocoPhillips	8.88	6.18	35.94	1.05	3.06	1	−10.38	−24.31
Continental Resources	11.11	2.5	4.76	1.72	0.72	1	−30.93	−54.08
Cosan Ltd	12.35	1.22	5.78	1.19	0.55	5	70.45	107.76
Crestwood Equity Partners LP	13.75	2.55	3.22	2.04	6.85	3	25.47	−5.25
...	...	...	...	...	...	...	...	...
Williams	327.38	0.07	8.6	1.56	6.68	1	3.13	−14.77

Table A.6 Evaluation Table input data for NYSE Financial Sector

Decision Matrix	P/E Ratio	EPS	Rev (B)	Beta	DY(%)	Mon	YTD (%)	1 Year
Nuveen CA MVF 2	43.47	0.38	0.002	0.06	3.09	5	8.01	4.77
Nuveen High Income 2020 Target Term	18.47	0.54	0.01	0.02	3.19	2	−0.4	1.54
Nuveen Dow 30Sm	15.72	1.12	0.015	0.97	6.72	3	9.12	−4.71
Ellsworth Growth Pref A	9.21	1.15	0.003	0.88	4.54	5	11.69	6.43
Federal Agricultural Mortgage A	9.16	8.94	0.609	1.19	3.42	5	26.23	10.68
Chimera Investment Pref A	21.89	0.91	1.36	0.62	10.06	5	6.26	5.08
Ares Management Pref A	39.19	0.68	1.35	1.47	4.64	5	4.65	5.18
Apollo Global Management A	30.02	1.29	1.72	1.49	5.12	5	60.76	29.9
Ladder Capital A	12.6	1.36	0.553	1.03	7.94	5	10.73	−2
Aberdeen Emerging Markets Equity	10.31	0.68	0.02	0.71	3.08	3	11.34	10.3
Aberdeen Asia-Pacific	71.33	0.06	0.083	0.51	7.8	3	9.3	5.49
Adams Diversified Equity Closed	7.95	1.99	0.031	1	1.27	5	25.67	1.28
Barclays ADR	9.14	0.91	26.52	0.97	4.28	1	3.9	−16.36
Santander Chile ADR	16.67	1.73	2.66	0.63	3.83	1	6.76	−11.69
Sumitomo Mitsui Financial ADR	7.37	0.94	24.33	1.23	9.39	3	−3.08	−4.48
Mitsubishi UFJ Financial ADR	8.26	0.61	29.95	1.37	3.94	1	10.74	−1.76
China Life Insurance ADR	14.81	0.81	101.14	1.54	0.97	1	−8.39	−29
Aegon ADR	10.19	0.41	49.17	1.36	7.94	1	14.68	11.91
Banco Bilbao ADR	6.65	0.79	24.59	1.06	5.52	1	0.38	−8.61
Credit Suisse ADR	13.51	0.91	18.39	1.44	2.09	1	13.44	−6.31
Prudential Public ADR	20.99	1.77	65.73	1.46	3.4	1	−62.49	−57.29
Lloyds Banking ADR	11.21	0.27	54.24	1.06	7.13	1	6.19	−6.06
ING ADR	8.35	1.3	24.65	1.35	6.98	1	5.03	−6.26

(continued)

Table A.6 (continued)

Decision Matrix	P/E Ratio	EPS	Rev (B)	Beta	DY(%)	Mon	YTD (%)	1 Year
BBVA Banco Frances ADR	2.69	1.58	1.45	0.73	6.13	2	20.31	5.84
Santander ADR	8.17	0.5	67.96	1.2	6.29	1	−4.91	−9.36
Itau CorpBanca ADR	18	0.63	1.56	0.97	1.98	1	−17.21	−23.76
Westpac Banking ADR	14.43	1.35	29.08	0.93	10.32	3	12.94	5.11
Nuveen California Div Advantag Muni	33.85	0.44	0.151	0.02	4.19	5	17.1	20.4
BlackRock Long Term Muni Advantage	15.71	0.83	0.012	0.19	4.64	5	21.56	19.98
Aflac	12.81	4.12	21.94	0.71	2.04	5	15.58	19.14
AG Mortgage Investment	26.24	0.57	0.163	0.95	12.08	1	−5.84	−12.89
AG Mortgage Invest Trust Pb Pref	26.16	0.57	0.163	0.95	12.12	3	4.75	−0.08
AG Mortgage Invest Trust Pa Pref	26.16	0.57	0.163	0.95	12.12	4	5.74	0.83
Federal Agricultural Mortgage	9.37	8.94	0.609	1.19	3.34	5	38.58	20.28
Great Ajax Corp	12.48	1.24	0.118	0.79	8.25	5	29.96	20.83
Alliance Data Systems	7.6	16.12	6.69	1.68	2.06	1	−17.9	−44.78
AllianceBernstein Holding LP	12.02	2.34	0.251	1.19	8.45	3	1.72	−6.96
AllianzGI Diversifiedome Convertibl	9.77	2.32	0.014	1.39	8.84	3	22.72	0.26
AllianzGI Equity Convertible Closed	17.27	1.22	0.018	1.11	7.19	3	14.56	−0.28
Ares Dynamic Credit Allocation Inc	22.69	0.65	0.044	0.55	8.76	1	7.46	−4.08
BlackRock Credit Allocationome Tr	16.86	0.8	0.117	0.34	7.43	5	20.75	13.45
Allstate	14.32	7.55	42.12	0.82	1.85	5	30.09	11.26
Ally Financial Inc	8.19	3.91	5.79	1.29	2.12	5	35.17	19.42
Artisan Partners AM	10.48	2.48	0.792	1.88	9.23	1	21.69	−5.49
New America High Income Closed Fund	10.24	0.89	0.02	0.55	7.28	5	19.18	10.3
Reinsurance of America	13.87	11.3	13.43	0.65	1.79	5	22.44	6.8
Bank of America	10.59	2.81	57.92	1.61	2.42	5	11.75	15.93
Nuveen Build America Bond Closed	30.63	0.71	0.04	0	5.35	5	12.86	12.4
American Financial	13.04	7.86	7.7	0.84	1.76	5	14.59	3.76
First American	12.52	4.76	5.76	0.88	2.82	5	32.82	27.53
...	...	...	...	...	...	...	...	...
Argo Group Int	20.6	3.29	1.91	0.58	1.83	5	1.58	12.93

Table A.7 Evaluation Table input data for NYSE Technology Sector

Decision Matrix	P/E Ratio	EPS	Rev (B)	Beta	DY(%)	Mon	YTD (%)	1 Year
ABB ADR	43.75	0.41	41.68	1.19	4.33	1	−4.52	−14.63
Accenture	25.05	7.36	43.22	1.05	1.74	5	30.37	15.3
SAP ADR	40.2	2.87	29.3	1.11	1.45	3	15.77	0.63
Infosys ADR	22.1	0.5	12.3	0.48	2.2	5	17.44	14.78
Wipro ADR	16.74	0.22	8.55	0.5	0.29	1	−5.71	−4.22
BT ADR	7.82	1.33	23.46	0.81	9.26	1	−30.46	−31.81
STMicroelectronics ADR	16.03	1.22	9.42	1.41	1.22	5	41.5	17.96
Canon ADR	16.66	1.58	33.8	0.59	5.61	1	−4.71	−14.78
Agilent Technologies	21.76	3.4	5.09	1.45	0.89	5	9.69	11.71
Allegion PLC	23.03	4.37	2.8	1.16	1.07	5	26.33	20.21
Ametek	25.01	3.52	5.04	1.21	0.64	5	29.91	20.81
Amphenol	23.8	4.01	8.33	1.04	1.05	5	17.88	12.33
AO Smith	18.66	2.48	3.08	1.49	1.9	1	8.52	−2.97
Scnc App In	29.82	2.8	5.58	1.31	1.77	5	30.96	17.34
Rockwell Automation	17.01	9.1	6.69	1.42	2.51	2	2.87	−10.41
AVX	9.68	1.6	1.74	1.17	2.98	2	1.38	−6.13
AZZ	18.11	2.13	0.953	1.43	1.76	1	−4.48	−15.61
Badger Meter	41.93	1.24	0.423	0.78	1.31	3	5.87	9.64
Belden	12.48	3.97	2.54	2.37	0.4	1	18.63	−18.62
Regal Beloit	12.62	5.47	3.53	1.6	1.74	1	−1.44	−9.69
Benchmark Electronics	23.01	1.28	2.5	0.85	2.04	5	38.67	26.32
Broadridge	30.08	4.07	4.36	0.7	1.76	5	27.31	4.42
BWX Tech	27.75	1.99	1.79	1.05	1.23	5	44.31	−4.07
CAE Inc.	26.88	0.91	2.58	0.82	1.25	5	32.81	29.22
Jabil Circuit	20.28	1.72	25.28	0.97	0.92	5	40.58	46.86
TE Connectivity	9.37	9.4	13.15	1.17	2.09	4	16.45	11.61
Issuer Direct Corp	61.59	0.17	0.015	0.76	1.95	1	−9.43	−31.92
CTS Corp	20.72	1.51	0.477	1.29	0.51	5	21.05	3.67
Danaher	40.74	3.41	20.25	0.95	0.49	5	34.56	34.22
Deluxe	19.87	2.27	2.01	1.35	2.66	1	17.33	−11.48
DXC Technology	6.7	3.94	20.36	1.96	3.18	1	−49.99	−69.67
Eaton	15.29	5.14	21.71	1.42	3.61	3	14.48	−1.47
Espey Mfg&Electronics	23.47	0.98	0.036	0.23	4.35	1	−7.7	−20.14
Methode Electronics	13.47	2.43	1.05	1.45	1.35	5	40.23	13.13
Emerson	20.23	3.14	18.29	1.37	3.08	3	8.87	−9.55
Energizer	91.94	0.43	2.23	0.67	3.05	1	−12.76	−33.7
Enersys	17.55	3.47	2.92	1.6	1.15	1	−21.48	−23.75
ESCO Technologies	24.93	3.24	0.807	1.07	0.4	5	22.44	32.4
Evertec Inc	23.81	1.31	0.471	0.73	0.64	5	8.4	35.67
FactSet Research	26.55	9.09	1.44	0.95	1.19	4	20.64	13.42

(continued)

Table A.7 (continued)

Decision Matrix	P/E Ratio	EPS	Rev (B)	Beta	DY(%)	Mon	YTD (%)	1 Year
Fortive	36.69	1.77	6.31	1.22	0.43	1	−3.3	−17.39
GlobalSCAPE	21.23	0.55	0.037	0.57	0.52	5	174.9	204.11
Northrop Grumman	18.15	20.26	32.89	0.8	1.44	5	50.57	21.95
Hewlett Packard	19.41	0.73	29.87	1.63	3.19	1	6.81	−8.5
Hexcel	22.98	3.36	2.32	1.04	0.88	5	34.74	26.68
Hill-Rom	32.46	3.02	2.88	0.86	0.86	5	10.71	11.13
HP Inc	5.89	2.72	58.72	1.47	4	1	−21.65	−32.28
Hubbell	19.84	6.63	4.61	1.48	2.56	5	32.37	8.23
IBM	11.75	12.01	77.86	1.36	4.59	4	24.16	0.2
Vishay Intertechnology	8.32	2.09	2.99	1.51	2.18	3	−3.33	−5.12
...	...	...	...	...	...	...	...	...
MSCI	33.64	6.62	1.48	1.12	1.22	5	51.02	44.54

Table A.8 Phase A results for NYSE Energy Sector

Name	ELECTRE 3	MAUT	PROMETHEE	TOPSIS
Petroleo Brasileiro ADR Reptg 2 Pref	7.69	39.55	14.15	48.63
Phillips 66	45.05	56.91	31.13	57.29
Phillips 66 Partners LP	38.22	51.22	30.92	52.48
Baker Hughes A	−33.91	23.96	−19.06	31.68
GasLog Partners Pref A	50.10	49.62	31.80	50.48
Adams Resources & Energy	−31.25	27.77	−14.89	41.21
Ecopetrol ADR	−10.05	31.04	−5.20	45.89
Total ADR	41.59	40.34	15.27	50.14
Petroleo Brasileiro Petrobras ADR	3.77	36.04	10.80	47.80
CNOOC ADR	46.46	44.38	3.98	54.57
Sinopec Shanghai Petrochemical ADR	19.84	32.62	−6.79	44.53
Royal Dutch Shell ADR	53.86	47.88	18.97	54.97
Equinor ADR	−15.52	33.15	2.12	45.97
ENI ADR	38.02	37.24	15.06	47.96
PetroChina ADR	48.68	43.36	5.50	51.66
Transportadora Gas ADR	40.38	34.41	−6.74	44.76
BP ADR	52.87	43.54	18.17	51.76
Royal Dutch Shell B ADR	53.58	47.59	17.83	54.60
Plains All American Pipeline	−0.31	35.05	5.02	47.65
YPF Sociedad Anonima	−46.86	25.51	−28.10	42.22

(continued)

Table A.8 (continued)

Archrock	−56.54	25.91	−7.47	45.01
Teck Resources B	−49.26	26.86	−29.55	43.31
BP Prudhoe Bay Royalty Trust	34.96	37.94	−5.74	44.78
Cabot Oil&Gas	−26.78	30.46	−11.82	43.73
Canadian Natural	−27.97	32.59	−3.75	46.99
Cenovus Energy Inc	−29.00	30.58	−2.69	44.43
Chevron	39.26	45.11	25.05	53.07
Cimarex Energy	−48.09	27.97	−32.74	43.53
CONSOL Coal	33.59	33.40	−2.31	45.33
Concho Resources	−54.79	23.49	−35.52	39.76
ConocoPhillips	−23.01	33.93	−7.10	47.14
Continental Resources	−59.52	23.08	−39.99	40.95
Cosan Ltd	30.66	55.74	22.84	57.70
Crestwood Equity Partners LP	−20.84	38.30	10.80	48.82
Crossamerica Partners LP	32.81	38.39	25.25	47.89
CVR Energy	35.79	49.90	26.43	52.54
Cypress Energy Partners LP	42.26	51.66	32.68	54.35
Delek Logistics Partners LP	28.54	39.40	22.82	48.59
Devon Energy	−64.52	23.77	−32.66	42.93
Encana	−59.82	21.64	−41.06	40.84
Holly Energy Partners LP	19.65	33.10	1.04	44.88
...	...	...	...	...
Williams	−60.82	17.77	−17.75	22.76

Table A.9 Phase A results for NYSE Technology Sector

Name	ELECTRE 3	MAUT	PROMETHEE	TOPSIS
ABB ADR	11.57	26.99	−22.32	30.96
Accenture	51.66	62.42	38.03	43.95
SAP ADR	1.74	40.47	−3.72	33.49
Infosys ADR	18.75	57.52	23.24	38.55
Wipro ADR	−28.89	23.38	−31.61	29.81
BT ADR	17.89	33.19	−19.79	34.52
STMicroelectronics ADR	10.58	54.31	16.04	39.92
Canon ADR	10.05	32.53	−10.75	34.49
Agilent Technologies	6.24	52.21	7.06	36.48
Allegion PLC	13.61	54.72	18.49	38.85
Ametek	12.99	53.93	16.71	38.90
Amphenol	12.00	54.99	16.37	37.65
AO Smith	−38.05	22.11	−34.04	30.28
Scnc App In	12.27	54.04	16.38	38.58
Rockwell Automation	−14.77	33.92	−13.80	33.17
AVX	−21.38	31.78	−19.73	31.76
AZZ	−41.62	20.76	−40.30	27.80
Badger Meter	−10.87	36.89	−12.37	30.58
Belden	−53.64	17.91	−41.09	29.68
Regal Beloit	−39.90	22.81	−33.96	30.32
Benchmark Electronics	18.31	56.26	24.59	40.38
Broadridge	16.30	56.31	21.43	37.98
BWX Tech	9.47	53.57	14.17	37.62
CAE Inc.	14.35	54.87	21.77	39.49
Jabil Circuit	25.64	58.72	33.93	43.90
TE Connectivity	19.82	52.45	10.45	39.97
Issuer Direct Corp	−42.43	18.35	−44.52	21.92
CTS Corp	4.02	51.20	5.86	35.71
Danaher	23.20	56.17	29.11	40.79
Deluxe	−35.54	23.27	−30.23	30.79
DXC Technology	−46.48	20.48	−38.87	25.97
...	...	...	...	...
Xerox	−3.92	48.70	4.03	40.19

Table A.10 Phase A results for NYSE Financial Sector

Name	ELECTRE 3	MAUT	PROMETHEE	TOPSIS
Nuveen CA MVF 2	38.46	46.75	1.96	28.30
Nuveen High Income 2020 Target Term	25.03	39.12	−7.67	29.03
Nuveen Dow 30Sm	5.26	39.64	−7.08	28.91
Ellsworth Growth Pref A	170.49	47.55	3.71	30.42
Federal Agricultural Mortgage A	117.01	48.83	16.19	33.06
Chimera Investment Pref A	−301.05	48.96	13.82	29.95
Ares Management Pref A	−100.73	40.85	−12.01	27.03
Apollo Global Management A	−206.36	49.72	14.39	35.24
Ladder Capital A	95.02	46.88	5.62	29.40
Aberdeen Emerging Markets Equity	125.01	41.73	−5.66	30.79
Aberdeen Asia-Pacific	−20.04	37.36	−8.76	26.31
Adams Diversified Equity Closed	163.72	46.79	0.19	31.29
Barclays ADR	−28.00	32.83	−20.10	28.61
Santander Chile ADR	−148.79	33.15	−17.59	27.93
Sumitomo Mitsui Financial ADR	62.70	40.39	−6.28	29.70
Mitsubishi UFJ Financial ADR	−42.05	33.11	−18.40	30.64
China Life Insurance ADR	−10.27	29.70	−32.27	32.63
Aegon ADR	−57.21	37.19	−0.50	33.61
Banco Bilbao ADR	74.45	33.46	−17.61	29.43
Credit Suisse ADR	−133.90	30.77	−25.57	29.27
Prudential Public ADR	−240.19	20.80	−35.21	23.81
Lloyds Banking ADR	−44.85	35.85	−5.50	32.11
ING ADR	−51.04	33.20	−15.32	30.23
BBVA Banco Frances ADR	361.38	41.06	0.82	31.81
Santander ADR	−64.06	34.57	−12.04	32.36
Itau CorpBanca ADR	−154.06	26.98	−38.60	24.33
Westpac Banking ADR	66.33	44.23	8.37	32.13
Nuveen California Div Advantag Muni	−133.58	50.80	14.67	30.98
BlackRock Long Term Muni Advantage	128.35	52.47	19.33	32.74
Aflac	109.97	50.37	16.05	33.31
AG Mortgage Investment	−242.42	32.11	−16.78	26.50
AG Mortgage Invest Trust Pb Pref	−326.60	40.67	−1.94	28.96
AG Mortgage Invest Trust Pa Pref	−364.20	43.98	1.55	29.05
Federal Agricultural Mortgage	131.64	50.89	23.55	34.86
Great Ajax Corp	140.96	52.22	23.22	33.46
Alliance Data Systems	46.65	25.47	−33.43	27.20
AllianceBernstein Holding LP	5.02	38.93	−8.49	28.66
AllianzGI Diversifiedome Convertibl	38.71	41.01	−1.25	30.88
AllianzGI Equity Convertible Closed	−52.78	39.95	−5.22	29.30
Ares Dynamic Credit Allocation Inc	−324.81	35.28	−5.84	28.43
BlackRock Credit Allocationome Tr	86.51	51.94	21.29	31.89
...	...	...	...	...
Bancroft	157.38	49.87	12.73	32.49

Table A.11 Statistical indexes for the selected securities

	MinRet	MaxRet	Median	Mean	SD	VaR99	VaR97	VaR95	Skewness	Kurtosis
NOC	−0.077808	0.063431	0.000676	0.000448	0.013467	−0.049242	−0.024005	−0.019069	−0.674012	6.010290
GSB	−0.145414	0.112329	0.000000	0.000529	0.025610	−0.068713	−0.049494	−0.037134	−0.178767	3.616340
ACN	−0.056337	0.051843	0.001509	0.000480	0.011213	−0.035905	−0.022927	−0.017429	−0.658652	3.198245
SNX	−0.106592	0.083184	0.001181	0.000057	0.018689	−0.062642	−0.037670	−0.027143	−0.647182	4.394566
IBM	−0.089540	0.071541	0.000000	−0.000162	0.012330	−0.036385	−0.022937	−0.017667	−0.478771	7.086411
TSM	−0.073449	0.052043	0.000766	0.000795	0.014376	−0.039892	−0.026994	−0.022551	−0.310216	1.943694
MSI	−0.054133	0.058159	0.001211	0.000757	0.012276	−0.035977	−0.026263	−0.018172	−0.169242	2.672691
JBL	−0.083417	0.134342	0.001056	0.000260	0.017974	−0.050843	−0.036011	−0.029023	0.238747	7.066522
ORCL	−0.107381	0.084395	0.000842	0.000388	0.013077	−0.039936	−0.027570	−0.019280	−0.578328	11.245113
MSCI	−0.084600	0.142090	0.001597	0.001079	0.015456	−0.039073	−0.028230	−0.021382	0.579821	12.724204
ROP	−0.083342	0.051727	0.000913	0.000553	0.012257	−0.034488	−0.022477	−0.018602	−0.770306	6.374014
DHR	−0.060621	0.063723	0.000638	0.000576	0.010544	−0.029102	−0.019503	−0.015026	0.017092	5.072333
LDOS	−0.234929	0.080472	0.000780	0.000090	0.017017	−0.050421	−0.026742	−0.022870	−3.663357	49.360685
BHE	−0.153169	0.056000	0.001546	0.000182	0.016489	−0.050639	−0.031140	−0.024771	−1.902171	13.712781
INFY	−0.079694	0.066125	0.000000	0.000287	0.014153	−0.036931	−0.027278	−0.021654	−0.212132	3.641736
HUBB	−0.119682	0.069599	0.000000	0.000088	0.014425	−0.035496	−0.026763	−0.021403	−0.945870	10.359222
NNI	−0.097416	0.127130	0.001553	0.000784	0.018295	−0.048459	−0.032050	−0.026833	0.186857	6.357661
CAE	−0.052734	0.057234	0.000978	0.000734	0.012595	−0.034766	−0.023228	−0.019422	−0.077775	2.497146
HXL	−0.063379	0.080656	0.000662	0.000406	0.014856	−0.038635	−0.027071	−0.022961	0.187992	3.901582
BR	−0.100575	0.089993	0.000853	0.000866	0.012592	−0.032041	−0.021722	−0.016875	−0.418587	10.824529
PSX	−0.052226	0.049867	0.000263	0.000156	0.013342	−0.035732	−0.029424	−0.022674	−0.069855	1.821956
NC	−0.088623	0.282276	0.002196	0.002056	0.028773	−0.058687	−0.050514	−0.040731	2.434129	22.467297
CELP	−0.207407	0.252144	0.000000	0.000172	0.038552	−0.091408	−0.071286	−0.060333	0.572526	6.099420

GLP	−0.206842	0.106250	0.000000	0.000207	0.023876	−0.053926	−0.039066	−0.034394	−0.404502	8.979767
SUN	−0.124734	0.152829	0.000000	−0.000267	0.021671	−0.054210	−0.036399	−0.030947	0.492752	9.187671
TRP	−0.044504	0.073667	0.000583	0.000225	0.012816	−0.031137	−0.024725	−0.020866	0.150190	1.782739
RDS-A	−0.079804	0.097415	0.000779	0.000454	0.015527	−0.037409	−0.028919	−0.024275	0.232366	4.328223
GLOP	−0.124204	0.118321	0.000432	0.000592	0.020525	−0.053575	−0.035926	−0.029338	0.029716	5.040865
PSXP	−0.076577	0.072812	0.000227	−0.000340	0.016672	−0.041726	−0.030632	−0.027324	0.064466	1.811138
RDS-B	−0.088053	0.102781	0.001063	0.000488	0.015950	−0.038365	−0.028660	−0.025406	0.231354	5.102997
INT	−0.191685	0.155319	0.000776	−0.000544	0.022117	−0.064801	−0.039957	−0.029993	−0.994850	16.387535
CZZ	−0.149343	0.129114	0.001208	0.001521	0.025560	−0.066787	−0.042760	−0.034030	0.060838	3.691457
MMP	−0.070588	0.062347	−0.000134	−0.000142	0.014212	−0.036271	−0.026411	−0.023050	0.204543	2.775524
CVI	−0.129289	0.217920	0.000450	0.000220	0.028745	−0.070141	−0.052375	−0.045590	0.747455	6.694582
BP	−0.078008	0.073658	0.000286	0.000404	0.015362	−0.040129	−0.028719	−0.024827	−0.112656	2.572632
CVX	−0.057246	0.071827	0.000537	0.000348	0.013054	−0.032956	−0.024475	−0.021098	−0.029018	2.861619
CEO	−0.066644	0.089765	0.000671	0.000682	0.017798	−0.047548	−0.033109	−0.027702	0.018867	1.796554
XOM	−0.057277	0.044560	0.000357	−0.000107	0.010948	−0.032464	−0.022010	−0.018559	−0.359791	2.129880
OKE	−0.102469	0.173115	0.001211	0.001221	0.020016	−0.051275	−0.034962	−0.029202	0.618880	9.150568
PTR	−0.060679	0.079934	−0.000577	0.000088	0.016750	−0.042951	−0.030406	−0.025048	0.368817	2.304897
RNR	−0.081077	0.074774	0.000495	0.000293	0.012492	−0.035414	−0.022093	−0.017910	−0.126635	6.468945
WTM	−0.047301	0.048832	0.000000	0.000263	0.009417	−0.022648	−0.016780	−0.014517	0.239646	3.256256
TPVG	−0.073724	0.075510	0.000000	0.000016	0.014773	−0.042986	−0.028417	−0.023696	−0.223974	3.070499
JPM	−0.059050	0.046361	0.000816	0.000650	0.013320	−0.037998	−0.025805	−0.020167	−0.246559	2.147910
RFI	−0.038693	0.034331	0.000000	−0.000151	0.008889	−0.026188	−0.017003	−0.015284	−0.351757	1.720516
SC	−0.176916	0.159692	0.000000	0.000474	0.025096	−0.071158	−0.043279	−0.034471	−0.018554	9.003217
BBN	−0.036272	0.042471	0.000437	0.000045	0.007102	−0.022376	−0.015489	−0.010964	−0.250472	4.240600
HIG	−0.065277	0.056965	0.000245	0.000123	0.012738	−0.035785	−0.025203	−0.021255	−0.105075	3.405590

(continued)

Table A.11 (continued)

	MinRet	MaxRet	Median	Mean	SD	VaR99	VaR97	VaR95	Skewness	Kurtosis
NVG	-0.030447	0.025381	0.000000	-0.000043	0.005344	-0.014984	-0.011292	-0.008829	-0.469559	3.911184
WFC	-0.101485	0.056119	0.000000	-0.000097	0.013671	-0.039483	-0.024013	-0.021315	-0.530029	5.006321
SAR	-0.084481	0.069110	0.000411	0.000484	0.015293	-0.040435	-0.030324	-0.023445	-0.231054	3.138825
ALL	-0.054447	0.061405	0.000770	0.000454	0.010449	-0.030831	-0.020749	-0.015893	-0.258645	4.973240
BTT	-0.036693	0.032045	0.000000	-0.000063	0.005571	-0.014312	-0.010154	-0.008074	-0.183475	6.377144
PMT	-0.120473	0.061785	0.000973	0.000395	0.013679	-0.040792	-0.027168	-0.022181	-1.322685	10.234292
DFP	-0.034714	0.035406	0.000372	-0.000106	0.007421	-0.021357	-0.016318	-0.012610	-0.350134	2.451549
MET	-0.080176	0.085203	0.000435	0.000093	0.016102	-0.050673	-0.034506	-0.026505	-0.078583	4.120069
PNC	-0.059738	0.045776	0.000617	0.000367	0.013296	-0.041085	-0.026321	-0.021236	-0.485273	2.093130
RGA	-0.056277	0.053854	0.000842	0.000743	0.012492	-0.037142	-0.024446	-0.019967	-0.196426	2.613224

References

1. Abdelaziz, F Ben, El Fayedh, Rimeh, and Rao, Ananth (2009). "A discrete stochastic goal program for portfolio selection: The case of United Arab Emirates equity market". In: *INFOR: Information Systems and Operational Research* 47.1, pp. 5–13.
2. Abdelaziz, Fouad Ben and Masmoudi, Meryem (2014). "A multiple objective stochastic portfolio selection problem with random Beta". In: *International Transactions in Operational Research* 21.6, pp. 919–933.
3. Ahmed, E and El-Alem, M (2005). "On multiobjective optimization in portfolio management". In: *Applied mathematics and computation* 167.1, pp. 616–621.
4. Akbay, Mehmet Anil, Kalayci, Can B, and Polat, Olcay (2020). "A parallel variable neighborhood search algorithm with quadratic programming for cardinality constrained portfolio optimization". In: *Knowledge-Based Systems*, p. 105944.
5. Albadvi, Amir, Chaharsooghi, S Kamal, and Esfahanipour, Akbar (2007). "Decision making in stock trading: An application of PROMETHEE". In: *European journal of operational research* 177.2, pp. 673–683.
6. Alexander, Gordon J and Resnick, Bruce G (1985). "Using linear and goal programming to immunize bond portfolios". In: *Journal of Banking & Finance* 9.1, pp. 35–54.
7. Amiri, Maghsoud, Zandieh, Mostafa, Vahdani, Behnam, Soltani, R, and Roshanaei, V (2010). "An integrated eigenvector–DEA–TOPSIS methodology for portfolio risk evaluation in the FOREX spot market". In: *Expert Systems with Applications* 37.1, pp. 509–516.
8. Ammar, Elsaid El (2008). "On solutions of fuzzy random multiobjective quadratic programming with applications in portfolio problem". In: *Information Sciences* 178.2, pp. 468–484.
9. Aouni, Belaid, Doumpos, Michael, Pérez-Gladish, Blanca, and Steuer, Ralph E (2018). "On the increasing importance of multiple criteria decision aid methods for portfolio selection". In: *Journal of the Operational Research Society* 69.10, pp. 1525–1542.
10. Arrow, Kenneth J, Raynaud, Hervé, et al. (1986). "Social choice and multicriterion decision-making". In: *MIT Press Books* 1.
11. Bäck, Thomas, Fogel, David B, and Michalewicz, Zbigniew (1997). *Handbook of evolutionary computation*. CRC Press.
12. Bäck, Thomas and Schwefel, Hans-Paul (1993). "An overview of evolutionary algorithms for parameter optimization". In: *Evolutionary computation* 1.1, pp. 1–23.
13. Bagheri, Noushin (2019). "Deterministic goal programming approach for Islamic portfolio selection". In: *Operational Research*, pp. 1–13.
14. Ballestero, E (1998). "Approximating the optimum portfolio for an investor with particular preferences". In: *Journal of the Operational Research Society* 49.9, pp. 998–1000.

E. Sarmas et al., *Multicriteria Portfolio Construction with Python*, Springer Optimization and Its Applications 163, https://doi.org/10.1007/978-3-030-53743-2

15. Ballestero, Enrique, Bravo, Mila, Pérez-Gladish, Blanca, Arenas-Parra, Mar, and Pla-Santamaria, David (2012). “Socially responsible investment: A multicriteria approach to portfolio selection combining ethical and financial objectives”. In: *European Journal of Operational Research* 216.2, pp. 487–494.
16. Ballestero, Enrique and Garcia-Bernabeu, Ana (2012). “Portfolio selection with multiple time horizons: A mean variance—Stochastic goal programming approach”. In: *INFOR: Information Systems and Operational Research* 50.3, pp. 106–116.
17. Ballestero, Enrique, Pérez-Gladish, Blanca, Arenas-Parra, Mar, and Bilbao- Terol, Amelia (2009). “Selecting portfolios given multiple Eurostoxx-based uncertainty scenarios: A stochastic goal programming approach from fuzzy betas”. In: *INFOR: Information Systems and Operational Research* 47.1, pp. 59–70.
18. Behzadian, Majid, Otaghsara, S Khanmohammadi, Yazdani, Morteza, and Ignatius, Joshua (2012). “A state-of the-art survey of TOPSIS applications”. In: *Expert Systems with applications* 39.17, pp. 13051–13069.
19. Benninga, Simon (2014). *Financial modeling*. MIT press.
20. Bernard III, Taylor W and Keown, Arthur J (1978). “A goal programming application of capital project selection in the production area”. In: *AIIE Transactions* 10.1, pp. 52–57.
21. Bertsimas, Dimitris, Darnell, Christopher, and Soucy, Robert (1999). “Portfolio construction through mixed-integer programming at Grantham, Mayo, Van Otterloo and Company”. In: *Interfaces* 29.1, pp. 49–66.
22. Bilbao-Terol, Amelia, Arenas-Parra, Mar, and Cañal-Fernández, Verónica (2012). “Selection of socially responsible portfolios using goal programming and fuzzy technology”. In: *Information Sciences* 189, pp. 110–125.
23. Bilbao-Terol, Amelia, Arenas-Parra, Mar, Cañal-Fernández, Verónica, and Antomil-Ibias, José (2014). “Using TOPSIS for assessing the sustainability of government bond funds”. In: *Omega* 49, pp. 1–17.
24. Bilbao-Terol, Amelia, Arenas-Parra, Mar, Cañal-Fernández, Verónica, and Bilbao-Terol, Celia (2013). “Selection of socially responsible portfolios using hedonic prices”. In: *Journal of business ethics* 115.3, pp. 515–529.
25. Bilbao-Terol, Amelia, Arenas-Parra, Mar, Cañal-Fernández, Verónica, and Bilbao-Terol, Celia (2016a). “Multi-criteria decision making for choosing socially responsible investment within a behavioral portfolio theory framework: A new way of investing into a crisis environment”. In: *Annals of Operations Research* 247.2, pp. 549–580.
26. Bilbao-Terol, Amelia, Arenas-Parra, Mar, Cañal-Fernández, Verónica, and Jiménez, Mariano (2016b). “A sequential goal programming model with fuzzy hierarchies to sustainable and responsible portfolio selection problem”. In: *Journal of the Operational Research Society* 67.10, pp. 1259–1273.
27. Booth, G Geoffrey and Dash Jr, Gordon H (1977). “Bank Portfolio Management Using Nonlinear Two Stage Goal Programming”. In: *Financial Review* 12.1, pp. 59–69.
28. Bouri, Abdelfettah, Martel, Jean-Marc, and Chabchoub, Habib (2002). “A multi-criterion approach for selecting attractive portfolio”. In: *Journal of Multi-Criteria Decision Analysis* 11.4-5, pp. 269–277.
29. Branke, Jürgen, Scheckenbach, B, Stein, Michael, Deb, Kalyanmoy, and Schmeck, Hartmut (2009). “Portfolio optimization with an envelope-based multi-objective evolutionary algorithm”. In: *European Journal of Operational Research* 199.3, pp. 684–693.
30. Brans, Jean-Pierre, Vincke, Ph, and Mareschal, Bertrand (1986). “How to select and how to rank projects: The PROMETHEE method”. In: *European journal of operational research* 24.2, pp. 228–238.
31. Bravo, Mila, Pla-Santamaria, David, and Garcia-Bernabeu, Ana (2010). “Portfolio selection from multiple benchmarks: A goal programming approach to an actual case”. In: *Journal of Multi-Criteria Decision Analysis* 17.5-6, pp. 155–166.
32. Briec, Walter, Kerstens, Kristiaan, and Van de Woestyne, Ignace (2013). “Portfolio selection with skewness: A comparison of methods and a generalized one fund result”. In: *European Journal of Operational Research* 230.2, pp. 412–421.

33. Calvo, Clara, Ivorra, Carlos, and Liern, Vicente (2012). “On the computation of the efficient frontier of the portfolio selection problem”. In: *Journal of Applied Mathematics* 2012.
34. Calvo, Clara, Ivorra, Carlos, and Liern, Vicente (2015). “Finding socially responsible portfolios close to conventional ones”. In: *International Review of Financial Analysis* 40, pp. 52–63.
35. Calvo, Clara, Ivorra, Carlos, and Liern, Vicente (2016). “Fuzzy portfolio selection with non-financial goals: exploring the efficient frontier”. In: *Annals of Operations Research* 245.1–2, pp. 31–46.
36. Calvo, Clara, Ivorra, Carlos, and Liern, Vicente (2018). “Controlling risk through diversification in portfolio selection with non-historical information”. In: *Journal of the Operational Research Society* 69.10, pp. 1543–1548.
37. Caplin, DA and Kornbluth, JSH (1975). “Multiobjective investment planning under uncertainty”. In: *Omega* 3.4, pp. 423–441.
38. Charnes, Abraham, Cooper, William W, and Ferguson, Robert O (1955). “Optimal estimation of executive compensation by linear programming”. In: *Management science* 1.2, pp. 138–151.
39. Charnes, Abraham and Cooper, William Wager (1957). “Management models and industrial applications of linear programming”. In: *Management Science* 4.1, pp. 38–91.
40. Chen, Chen-Tung and Hung, Wei-Zhan (2009). “A new decision-making method for stock portfolio selection based on computing with linguistic assessment”. In: *Advances in Decision Sciences* 2009.
41. Chen, S (1995). “Multiobjective decision making on the portfolio selection”. In: *Proceeding of the 6th National Conference on Multiple Criteria Decision Making, Beijing, SCI-TECH Information Services, Windsor, England*, pp. 169–172.
42. Cheung, Michael Tow and Liao, Ziqi (2009). “Investing in real-world equity markets with an AHP-based decision framework”. In: *Journal of decision systems* 18.2, pp. 149–163.
43. Chunhachinda, Pornchai, Dandapani, Krishnan, Hamid, Shahid, and Prakash, Arun J (1997). “Portfolio selection and skewness: Evidence from international stock markets”. In: *Journal of Banking & Finance* 21.2, pp. 143–167.
44. Coello, Carlos A Coello, Lamont, Gary B, Van Veldhuizen, David A, et al. (2007). *Evolutionary algorithms for solving multi-objective problems* Vol. 5. Springer.
45. Coffin, Mark A and Taylor III, Bernard W (1996). “Multiple criteria R&D project selection and scheduling using fuzzy logic”. In: *Computers & Operations Research* 23.3, pp. 207–220.
46. Colson, Gerard and De Bruyn, Christian (1989). “An integrated multiobjective portfolio management system”. In: *Models and Methods in Multiple Criteria Decision Making*. Elsevier, pp. 1359–1381.
47. Cooper, WW, Lelas, V, and Sueyoshi, T (1997). “Goal programming models and their duality relations for use in evaluating security portfolio and regression relations”. In: *European Journal of Operational Research* 98.2, pp. 431–443.
48. Costa, Carlos A Bana e and Vansnick, Jean-Claude (1994). “MACBETH—An interactive path towards the construction of cardinal value functions”. In: *International transactions in operational Research* 1.4, pp. 489–500.
49. Dantzig, George Bernard (1998). *Linear programming and extensions*. Vol. 48. Princeton university press.
50. Dasgupta, Dipankar and Michalewicz, Zbigniew (2013). *Evolutionary algorithms in engineering applications*. Springer Science & Business Media.
51. Davies, Ryan J, Kat, Harry M, and Lu, Sa (2016). “Fund of hedge funds portfolio selection: A multiple-objective approach”. In: *Derivatives and Hedge Funds*. Springer, pp. 45–71.
52. De Jong, Kenneth (2016). “Evolutionary computation: a unified approach”. In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pp. 185–199.
53. De Keyser, Wim SM and Peeters, PHM (1994). “ARGUS—a new multiple criteria method based on the general idea of outranking”. In: *Applying multiple criteria aid for decision to environmental management*. Springer, pp. 263–278.

54. Devaud, JM, Groussaud, G, and Jacquet-Lagreze, Eric (1980). "UTADIS: Une méthode de construction de fonctions d'utilité additives rendant compte de jugements globaux". In: *European Working Group on Multicriteria Decision Aid, Bochum* 94.
55. Dominiak, Cezary (1997). "An application of interactive multiple goal programming on the Warsaw stock exchange". In: *Advances in multiple objective and goal programming*. Springer, pp. 66–74.
56. Doumpos, Michael (2009). *Multicriteria decision support systems. Lecture notes. Crete*.
57. Doumpos, Michael, Spanos, M, and Zopounidis, Constantin (1999). "On the use of goal programming techniques in the assessment of financial risks". In: *Journal of Euro-Asian Management* 5.1, pp. 83–100.
58. Doumpos, Michael and Zopounidis, Constantin (2002). *Multicriteria decision aid classification methods*. Vol. 73. Springer Science & Business Media.
59. Ehrgott, Matthias, Klamroth, Kathrin, and Schwehm, Christian (2004). "An MCDM approach to portfolio optimization". In: *European Journal of Operational Research* 155.3, pp. 752–770.
60. Elton, Edwin J, Gruber, Martin J, Brown, Stephen J, and Goetzmann, William N (2014). *Modern portfolio theory and investment analysis*. John Wiley & Sons.
61. Fauzi, Nur Aqilah Mohammed, Ismail, Munira, Jaaman, Saiful Hafizah, and Kamaruddin, Siti Noor Diana Mohd (2019). "Applicability of TOPSIS Model and Markowitz Model". In: *Journal of Physics: Conference Series*. Vol. 1212. 1. IOP Publishing, p. 012032.
62. Fogel, David B (2006). *Evolutionary computation: toward a new philosophy of machine intelligence*. Vol. 1. John Wiley & Sons.
63. Garcia-Melón, Mónica, Pérez-Gladish, Blanca, Gómez-Navarro, Tomás, and Mendez-Rodriguez, Paz (2016). "Assessing mutual funds' corporate social responsibility: a multistakeholder-AHP based methodology". In: *Annals of Operations Research* 244.2, pp. 475–503.
64. Garcia, Fernando, Guijarro, Francisco, and Moya, Ismael (2013). "A multiobjective model for passive portfolio management: An application on the S&P 100 index". In: *Journal of Business Economics and Management* 14.4, pp. 758–775.
65. Ghahtarani, Alireza and Najafi, Amir Abbas (2013). "Robust goal programming for multi-objective portfolio selection problem". In: *Economic Modelling* 33, pp. 588–592.
66. Giarlotta, Alfio (2001). "Multicriteria compensability analysis". In: *European Journal of Operational Research* 133.1, pp. 190–209.
67. Gladish, B Perez, Jones, DF, Tamiz, Merdah, and Terol, A Bilbao (2007). "An interactive three-stage model for mutual funds portfolio selection". In: *Omega* 35.1, pp. 75–88.
68. Greco, Salvatore (1997). "A new PCCA method: IDRA". In: *European Journal of Operational Research* 98.3, pp. 587–601.
69. Gupta, Pankaj, Mehlawat, Mukesh Kumar, Yadav, Sanjay, and Kumar, Arun (2019). "A polynomial goal programming approach for intuitionistic fuzzy portfolio optimization using entropy and higher moments". In: *Applied Soft Computing* 85, p. 105781.
70. Gupta, Pankaj, Mittal, Garima, and Mehlawat, Mukesh Kumar (2013). "Multiobjective expected value model for portfolio selection in fuzzy environment". In: *Optimization Letters* 7.8, pp. 1765–1791.
71. Han, Bing, Zhang, Xiao-Xiong, and Yi, Yujiang (2019). "Multi-criteria project selection using fuzzy preference relations based AHP and TOPSIS". In: *2019 Chinese Control And Decision Conference (CCDC)*. IEEE, pp. 3809–3816.
72. Harrington, Thomas C and Fischer, William A (1980). "Portfolio modeling in multiple-criteria situations under uncertainty: Comment". In: *Decision Sciences* 11.1, pp. 171–177.
73. Hashemi, Shide Sadat, Hajiagha, Seyed Hossein Razavi, Zavadskas, Edmundas Kazimieras, and Mahdiraji, Hannan Amoozad (2016). "Multicriteria group decision making with ELECTRE III method based on interval-valued intuitionistic fuzzy information". In: *Applied mathematical modelling* 40.2, pp. 1554–1564.
74. Hasuike, Takashi and Katagiri, Hideki (2014). "Risk-controlled multiobjective portfolio selection problem using a principle of compromise". In: *Mathematical Problems in Engineering* 2014.

75. Hinloopen, Edwin, Nijkamp, Peter, and Rietveld, Piet (1983). “Qualitative discrete multiple criteria choice models in regional planning”. In: *Regional Science and Urban Economics* 13.1, pp. 77–102.
76. Hokkanen, J., Salminen, P., (1997). *Choosing a solid waste management system using multicriteria decision analysis*. European Journal of Operational Research, 98(1), 19–36.
77. Holland, John H (1962). “Outline for a logical theory of adaptive systems”. In: *Journal of the ACM (JACM)* 9.3, pp. 297–314.
78. Horn, Jeffrey, Nafpliotis, Nicholas, and Goldberg, David E (1994). “A niched Pareto genetic algorithm for multiobjective optimization”. In: *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*. IEEE, pp. 82–87.
79. Hunter, John D (2007). “Matplotlib: A 2D graphics environment”. In: *Computing in science & engineering* 9.3, pp. 90–95.
80. Hurson, Ch and Zopounidis, Constantin (1997). “On the use of multicriteria decision aid methods to portfolio selection”. In: *Multicriteria analysis*. Springer, pp. 496–507.
81. Hurson, Christian and Zopounidis, Constantin (1996). *Gestion de portefeuille et analyse multicritere*. FeniXX.
82. Hwang, Ching-Lai and Yoon, Kwangsun (1981). “Methods for multiple attribute decision making”. In: *Multiple attribute decision making*. Springer, pp. 58–191.
83. Jacquet-Lagreze, Eric (1995). “An application of the UTA discriminant model for the evaluation of R & D projects”. In: *Advances in multicriteria analysis*. Springer, pp. 203–211.
84. Jacquet-Lagreze, Eric and Siskos, Jean (1982). “Assessing a set of additive utility functions for multicriteria decision-making, the UTA method”. In: *European journal of operational research* 10.2, pp. 151–164.
85. Jacquet-Lagrèze, Eric and Siskos, Jean (1983). *Méthode de décision multicritère*. Ed. Hommes et techniques.
86. Jacquet-Lagreze, Eric and Siskos, Yannis (2001). “Preference disaggregation: 20 years of MCDA experience”. In: *European Journal of Operational Research* 130.2, pp. 233–245.
87. Jayasekara, Pubudu LW, Adelgren, Nathan, and Wiecek, Margaret M (n.d.). “On convex multiobjective programs with application to portfolio optimization”. In: *Journal of Multi-Criteria Decision Analysis* .
88. Julong, Deng (1989). “Introduction to grey system theory”. In: *The Journal of grey system* 1.1, pp. 1–24.
89. Keeney, Ralph L, Raiffa, Howard, et al. (1993). *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press.
90. Khan, Ameer Hamza, Cao, Xinwei, Katsikis, Vasilios N, Stanimirović, Predrag, Brajević, Ivona, Li, Shuai, Kadry, Seifedine, and Nam, Yunyoung (2020). “Optimal Portfolio Management for Engineering Problems Using Nonconvex Cardinality Constraint: A Computing Perspective”. In: *IEEE Access* 8, pp. 57437–57450.
91. Khorramshahgol, Reza and Okoruwa, A Ason (1994). “A goal programming approach to investment decisions: A case study of fund allocation among different shopping malls”. In: *European Journal of Operational Research* 73.1, pp. 17–22.
92. Khoury, Nabil, Martel, Jean-Marc, and Veilleux, Marc (1993). “Méthode multicritère de sélection de portefeuilles indiciels internationaux”. In: *L’Actualité économique* 69.1, pp. 171–190.
93. K100r100ş, Şafak and Ustun, Ozden (2012). “An integrated approach for stock evaluation and portfolio optimization”. In: *Optimization* 61.4, pp. 423–441.
94. Kobayashi, Kiyoshi, Tanino, Tetuzo, Wierzbicki, Andrzej P, and Okumura, Makoto (1987). “Multifactor decision analysis for regional investment allocation”. In: *Toward Interactive and Intelligent Decision Support Systems*. Springer, pp. 422–431.
95. Kocadağl100, Ozan and Keskin, R100dvan (2015). “A novel portfolio selection model based on fuzzy goal programming with different importance and priorities”. In: *Expert Systems with Applications* 42.20, pp. 6898–6912.

96. Konno, Hiroshi, Shirakawa, Hiroshi, and Yamazaki, Hiroaki (1993). "A mean-absolute deviation-skewness portfolio optimization model". In: *Annals of Operations Research* 45.1, pp. 205–220.
97. Konno, Hiroshi and Suzuki, Ken-ichi (1995). "A mean-variance-skewness portfolio optimization model". In: *Journal of the Operations Research Society of Japan* 38.2, pp. 173–187.
98. Kumar, PC and Philippatos, GC (1979). "Conflict resolution in investment decisions: Implementation of goal programming methodology for dual- purpose funds". In: *Decision Sciences* 10.4, pp. 562–576.
99. Kumar, PC, Philippatos, George C, and Ezzell, John R (1978). "Goal Programming and the Selection of Portfolios by Dual-Purpose Funds". In: *The Journal of Finance* 33.1, pp. 303–310.
100. L'Hoir, H and Teghem, J (1995). "Portfolio selection by MOLP using an interactive branch and bound". In: *Foundations of Computing and Decision Sciences* 20.3, pp. 175–185.
101. Lamata, Maria Teresa, Liern, Vicente, and Pérez-Gladish, Blanca (2018). "Doing good by doing well: A MCDM framework for evaluating corporate social responsibility attractiveness". In: *Annals of Operations Research* 267.1–2, pp. 249–266.
102. Leão, Heloise Acco Tives, Canedo, Edna Dias, Costa, Pedro Henrique Teixeira, Okimoto, Márcio Vinicius, and Santos, Giovanni Almeida (2019). "Use of AHP and Promethee for Research Project Portfolio Selection". In: *International Conference on Computational Science and Its Applications*. Springer, pp. 504–517.
103. Leclercq, J (1984). "Propositions d'extension de la notion de dominance en présence de relations d'ordre sur les pseudo-critéres: la méthode MELCHIOR". In: *JORBEL-Belgian Journal of Operations Research, Statistics, and Computer Science* 24.1, pp. 32–46.
104. Lee, Sang M and Chesser, Delton L (1980). "Goal programming for portfolio selection". In: *The Journal of Portfolio Management* 6.3, pp. 22–26.
105. Lee, Sang M and Lerro, AJ (1973). "Optimizing the portfolio selection for mutual funds". In: *The Journal of Finance* 28.5, pp. 1087–1101.
106. Leung, Mark T, Daouk, Hazem, and Chen, An-Sing (2001). "Using investment portfolio return to combine forecasts: a multiobjective approach". In: *European Journal of Operational Research* 134.1, pp. 84–102.
107. Lipowski, Adam and Lipowska, Dorota (2012). "Roulette-wheel selection via stochastic acceptance". In: *Physica A: Statistical Mechanics and its Applications* 391.6, pp. 2193–2196.
108. Liu, Yong-Jun, Zhang, Wei-Guo, and Xu, Wei-Jun (2012). "Fuzzy multi-period portfolio selection optimization models using multiple criteria". In: *Automatica* 48.12, pp. 3042–3053.
109. MacCrimmon, Kenneth R (1968). *Decision making among multiple-attribute alternatives: a survey and consolidated approach*. Tech. rep. Rand Corp Santa Monica Ca.
110. Mansini, Renata and Speranza, Maria Grazia (1999). "Heuristic algorithms for the portfolio selection problem with minimum transaction lots". In: *European Journal of Operational Research* 114.2, pp. 219–233.
111. Marasović, Branka and Babić, Zoran (2011). "Two-step multi-criteria model for selecting optimal portfolio". In: *International Journal of Production Economics* 134.1, pp. 58–66.
112. Markowitz, Harry (1952). "Portfolio Selection". In: *The Journal of Finance* 7.1, pp. 77–91.
113. Markowitz, Harry (1959). *Portfolio selection: Efficient diversification of investments*. Vol. 16. John Wiley New York.
114. Marqués, Ana I, Garc100éa, Vicente, and Sánchez, J Salvador (n.d.). "Ranking-based MCDM models in financial management applications: analysis and emerging challenges". In:.
115. Martel, J-M, Khoury, NT, and Bergeron, M (1988). "An application of a multicriteria approach to portfolio comparisons". In: *Journal of the Operational Research Society* 39.7, pp. 617–628.
116. Martel, Jean-Marc, Khoury, Nabil, and M'Zali, Bouchra (1991). "Comparaison performance-taille des fonds mutuels par une analyse multicritère". In: *L'Actualité économique* 67.3, pp. 306–324.
117. Masmoudi, Meryem and Abdelaziz, Fouad Ben (2017). "A chance constrained recourse approach for the portfolio selection problem". In: *Annals of Operations Research* 251.1-2, pp. 243–254.

118. Masri, Hatem (2018). “A Shariah-compliant portfolio selection model”. In: *Journal of the Operational Research Society* 69.10, pp. 1568–1575.
119. Matarazzo, Benedetto (1988). “Preference ranking global frequencies in multicriterion analysis (PRAGMA)”. In: *European Journal of Operational Research* 36.1, pp. 36–49.
120. Matarazzo, Benedetto (1990). “A pairwise criterion comparison approach: the MAPPAC and PRAGMA methods”. In: *Readings in multiple criteria decision aid*. Springer, pp. 253–273.
121. Mavrotas, George, aas, Panos, and Psarras, John (2008). “An integrated multiple criteria methodology for supporting common stock portfolio selection decisions”. In: *Proceedings of the 67th meeting of the European working group on multiple criteria decision aiding. Rovaniemi, Finland*, pp. 56–71.
122. McKinney, Wes et al. (2010). “Data structures for statistical computing in python”. In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX, pp. 51–56.
123. Mehlawat, Mukesh Kumar (2016). “Credibilistic mean-entropy models for multi-period portfolio selection with multi-choice aspiration levels”. In: *Information Sciences* 345, pp. 9–26.
124. Messaoudi, Laila, Aouni, Belaid, and Rebai, Abdelwaheb (2017). “Fuzzy chance-constrained goal programming model for multi-attribute financial portfolio selection”. In: *Annals of Operations Research* 251.1–2, pp. 193–204.
125. Moon, Yongma and Yao, Tao (2011). “A robust mean absolute deviation model for portfolio optimization”. In: *Computers & Operations Research* 38.9, pp. 1251–1258.
126. Mousseau, V., (1995). *Eliciting information concerning the relative importance of criteria*. In: Pardalos, P., Siskos, Y., Zopounidis, C., editors. Advances in multicriteria analysis. Dordrecht: Kluwer Academic Publishers, 17–43.
127. Muhlemann, Alan Paul and Lockett, Alan Geoffrey (1980). “Portfolio modeling in multiple-criteria situations under uncertainty: rejoinder”. In: *Decision Sciences* 11.1, pp. 178–180.
128. Muhlemann, Alan Paul, Lockett, Alan Geoffrey, and Gear, Anthony Edward (1978). “Portfolio modeling in multiple-criteria situations under uncertainty”. In: *Decision Sciences* 9.4, pp. 612–626.
129. Nakayama, H, Takeguchi, T, and Sano, M (1983). “Interactive graphics for portfolio selection”. In: *Essays and Surveys on Multiple Criteria Decision Making*. Springer, pp. 280–289.
130. Nguyen, Thi Thanh and Gordon-Brown, Lee (2011). “Constrained fuzzy hierarchical analysis for portfolio selection under higher moments”. In: *IEEE Transactions on Fuzzy Systems* 20.4, pp. 666–682.
131. Nijkamp, P, Rietveld, P, and Voogd, H (1990). *Multiple criteria analysis in physical planning*.
132. Ogryczak, Wlodzimierz (2002). “Multiple criteria optimization and decisions under risk”. In: *Control and Cybernetics* 31.4, pp. 975–1003.
133. Ogryczak, Wlodzimierz (2000). “Multiple criteria linear programming model for portfolio selection”. In: *Annals of Operations Research* 97.1-4, pp. 143–162.
134. Oliphant, Travis E (2006). *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.
135. Opricovic, Serafim and Tzeng, Gwo-Hshiung (2004). “Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS”. In: *European journal of operational research* 156.2, pp. 445–455.
136. Opricovic, Serafim and Tzeng, Gwo-Hshiung (2007). “Extended VIKOR method in comparison with outranking methods”. In: *European journal of operational research* 178.2, pp. 514–529.
137. Orne, Daniel L, Rao, Ashok, and Wallace, William A (1975). “Profit maximization with the aid of goal programming for speculative housing estate developers”. In: *Journal of the Operational Research Society* 26.4, pp. 813–826.
138. Paelinck, Jean (1977). “Qualitative multicriteria analysis: an application to airport location”. In: *Environment and Planning A* 9.8, pp. 883–895.
139. Paelinck, Jean HP (1976). “Qualitative multiple criteria analysis, environmental protection and multiregional development”. In: *Papers of the Regional Science Association*. Vol. 36. 1. Springer, pp. 59–74.

140. Pardalos, Panos M, Sandström, Mattias, and Zopounidis, Costas (1994). "On the use of optimization models for portfolio selection: A review and some computational results". In: *Computational Economics* 7.4, pp. 227–244.
141. Parra, M Arenas, Terol, A Bilbao, and Ur100a, MV Rodr100guez (2001). "A fuzzy goal programming approach to portfolio selection". In: *European Journal of Operational Research* 133.2, pp. 287–297.
142. Pätäri, Eero, Karell, Ville, Luukka, Pasi, and Yeomans, Julian S (2018). "Comparison of the multicriteria decision-making methods for equity portfolio selection: The US evidence". In: *European Journal of Operational Research* 265.2, pp. 655–672.
143. Pendaraki, Konstantina, Zopounidis, Constantin, and Doumpos, Michael (2005). "On the construction of mutual fund portfolios: A multicriteria methodology and an application to the Greek market of equity mutual funds". In: *European Journal of Operational Research* 163.2, pp. 462–481.
144. Perez-Gladish, Blanca and M'Zali, Bouchra (2010). "An AHP-based approach to mutual funds' social performance measurement". In: *International Journal of Multicriteria Decision Making* 1.1, pp. 103–127.
145. Petrillo, Antonella, De Felice, Fabio, Garc100éa-Melón, Mónica, and Pérez-Gladish, Blanca (2016). "Investing in socially responsible mutual funds: Proposal of non-financial ranking in Italian market". In: *Research in International Business and Finance* 37, pp. 541–555.
146. Poon, Pui Wah and Carter, Jonathan Neil (1995). "Genetic algorithm crossover operators for ordering applications". In: *Computers & Operations Research* 22.1, pp. 135–147.
147. Powell, John G and Premachandra, IM (1998). "Accommodating diverse institutional investment objectives and constraints using non-linear goal programming". In: *European Journal of Operational Research* 105.3, pp. 447–456.
148. Prakash, Arun J, Chang, Chun-Hao, and Pactwa, Therese E (2003). "Selecting a portfolio with skewness: Recent evidence from US, European, and Latin American equity markets". In: *Journal of Banking & Finance* 27.7, pp. 1375–1390.
149. Qi, Yue (2019). "Parametrically computing efficient frontiers of portfolio selection and reporting and utilizing the piecewise-segment structure". In: *Journal of the Operational Research Society*, pp. 1–16.
150. R100éos-Garcĭ00a, S and Rios-Insua, S (1983). "The portfolio selection problem with multiattributes and multiple criteria". In: *Essays and Surveys on Multiple Criteria Decision Making*. Springer, pp. 317–325.
151. Rogers, M., Bruen, M., (1998). *A new system for weighting environmental criteria for use within ELECTRE III*. European Journal of Operational Research, 107(3), 552–63.
152. Roman, Diana, Darby-Dowman, Kenneth, and Mitra, Gautam (2007). "Mean-risk models using two risk measures: a multi-objective approach". In: *Quantitative Finance* 7.4, pp. 443–458.
153. Roubens, Marc (1982). "Preference relations on actions and criteria in multicriteria decision making". In: *European Journal of Operational Research* 10.1, pp. 51–55.
154. Roy, B (1991). *The outranking approach and the foundations of ELECTRE methods. Theory and decisions 31 (1): 49–73*.
155. Roy, B (1996). "Multicriteria methodology for decision aiding. Kluwer Academic Publishers". In: *Dordrech, Netherlands*.
156. Roy, Bernard (1968). "Classement et choix en présence de points de vue multiples". In: *Revue française d'informatique et de recherche opérationnelle* 2.8, pp. 57–75.
157. Roy, Bernard (1985). *Méthodologie multicritère d'aide à la décision*. Economica.
158. Rustem, Berc and Rustem, B (1998). *Algorithms for nonlinear programming and multiple-objective decisions*. Wiley Chichester.
159. Rys, Tomasz and Ziemba, W (1991). "Multiobjective investment scheduling problem". In: *Lecture Notes in Economics and Mathematical Systems* 351, pp. 245–253.
160. Saaty, Thomas L (2001). "Decision making with the analytic network process (ANP) and its super decisions software: the national missile defense (NMD) example". In: *ISAHP 2001 proceedings*, pp. 2–4.

161. Saaty, TL (1980). “The Analytic Hierarchy Process”. In: *Mc Graw-Hill*.
162. Schlottmann, Frank and Seese, Detlef (2004). “A hybrid heuristic approach to discrete multi-objective optimization of credit portfolios”. In: *Computational statistics & data analysis* 47.2, pp. 373–399.
163. Schmitt, Lothar M (2001). “Theory of genetic algorithms”. In: *Theoretical Computer Science* 259.1-2, pp. 1–61.
164. Schniederjans, AM and Schniederjans, MJ (2000). “Using the analytic hierarchy process and multi-objective programming for internet portfolio analysis”. In: *Multi-Criteria Applications, JAI/Elsevier, Amsterdam*, pp. 103–137.
165. Schniederjans, Marc J (1984). *Linear goal programming*. Petrocelli Books.
166. Schniederjans, Marc J, Zorn, Thomas S, and Johnson, Robert R (1993). “Allocating total wealth: A goal programming approach”. In: *Computers & operations research* 20.7, pp. 679–685.
167. Sealey Jr, CW (1977). “Commercial bank portfolio management with multiple objectives”. In: *Journal of Commercial Bank Lending* 59.6, pp. 39–48.
168. Sevastjanov, Pavel and Dymova, Ludmila (2009). “Stock screening with use of multiple criteria decision making and optimization”. In: *Omega* 37.3, pp. 659–671.
169. Shapiro, Jeremy F (1976). “Multiple criteria public investment decision making by mixed integer programming”. In: *Multiple Criteria Decision Making*. Springer, pp. 170–182.
170. Sheshai, KM El, Harwood, GB, and Hermanson, RH (1977). “Cost volume profit analysis with integer goal programming”. In: *Management Accounting* 59.4, pp. 43–47.
171. Shing, Chue and Nagasawa, Hiroyuki (1999). “Interactive decision system in stochastic multiobjective portfolio selection”. In: *International Journal of Production Economics* 60, pp. 187–193.
172. Simos, J., (1990). *L’evaluation environnementale: Un processus cognitif negocie*. These de doctorat, DGF-EPFL, Lausanne.
173. Siskos, Yannis and Yannacopoulos, Denis (1985). “UTASTAR: An ordinal regression method for building additive value functions”. In: *Investigaçao Operacional* 5.1, pp. 39–53.
174. Skocz, Maciej, Zebrowski, Maciej, and Ziembla, Wieslaw (1989). “Spatial allocation and investment scheduling in the development programming”. In: *Aspiration Based Decision Support Systems*. Springer, pp. 322–338.
175. Skulimowski, Andrzej MJ (1996). “Dynamic portfolio optimization based on reference trajectories”. In: *System Modelling and Optimization*. Springer, pp. 171–178.
176. Speranza, M Grazia (1993). *Linear programming models for portfolio optimization*.
177. Speranza, M Grazia (1996). “A heuristic algorithm for a portfolio optimization model applied to the Milan stock market”. In: *Computers & Operations Research* 23.5, pp. 433–441.
178. Speranza, Maria Grazia (1994). “Linear models for portfolio selection and their application to the Milano stock market”. In: *Financial modelling*. Springer, pp. 320–333.
179. Spronk, Jaap, Steuer, Ralph E, and Zopounidis, Constantin (2005). “Multicriteria decision aid/analysis in finance”. In: *Multiple criteria decision analysis: State of the art surveys*. Springer, pp. 799–848.
180. Spronk, Jaap and Veeneklaas, Frank (1983). “A feasibility study of economic and environmental scenarios by means of interactive multiple goal programming”. In: *Regional Science and Urban Economics* 13.1, pp. 141–160.
181. Spronk, Jaap and Zambruno, G (1985). “Interactive multiple goal programming for bank portfolio selection”. In: *Multiple Criteria Decision Methods and Applications*. Springer, pp. 289–306.
182. Spronk, Jaap and Zambruno, Giovanni M (1981). *A multiple-criteria approach to portfolio selection*.
183. Steuer, Ralph E (1986). “Multiple criteria optimization”. In: *Theory, computation and applications*.
184. Steuer, Ralph E and Na, Paul (2003). “Multiple criteria decision making combined with finance: A categorized bibliographic study”. In: *European Journal of operational research* 150.3, pp. 496–515.

185. Steuer, Ralph E, Qi, Yue, and Hirschberger, Markus (2005). "Multiple objectives in portfolio selection". In: *Journal of Financial Decision Making* 1.1, pp. 5–20.
186. Steuer, Ralph E, Qi, Yue, and Hirschberger, Markus (2006). "Developments in multi-attribute portfolio selection". In: *Multiple criteria decision making* 5, pp. 251–262.
187. Steuer, Ralph E, Qi, Yue, and Hirschberger, Markus (2007). "Suitable-portfolio investors, nondominated frontier sensitivity, and the effect of multiple objectives on standard portfolio selection". In: *Annals of Operations Research* 152.1, pp. 297–317.
188. Steuer, Ralph E, Qi, Yue, and Hirschberger, Markus (2008a). "Portfolio selection in the presence of multiple criteria". In: *Handbook of financial engineering*. Springer, pp. 3–24.
189. Steuer, Ralph E, Qi, Yue, and Hirschberger, Markus (2008b). "Portfolio selection in the presence of multiple criteria". In: *Handbook of financial engineering*. Springer, pp. 3–24.
190. Steuer, RE (1974). "Interval criterion weights programming: A portfolio selection example, gradient cone modification, and computational experience". In: *Proceedings: Tenth annual southeastern regional TIMS meeting*, pp. 246–255.
191. Stone, Bernell K (1973). "A linear programming formulation of the general portfolio selection problem". In: *Journal of Financial and Quantitative Analysis* 8.4, pp. 621–636.
192. Stone, Bernell K and Reback, Robert (1975). "Constructing a model for managing portfolio revisions". In: *Journal of Bank Research* 6.1, pp. 48–60.
193. Storn, Rainer and Price, Kenneth (1997). "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces". In: *Journal of global optimization* 11.4, pp. 341–359.
194. Sugrue, Paul K, Mehrotra, Anuj, and Orehovec, Paul M (2006). "Financial aid management: an optimisation approach". In: *International Journal of Operational Research* 1.3, pp. 267–282.
195. Tamiz, Mehrdad and Azmi, Rania A (2019). "Goal programming with extended factors for portfolio selection". In: *International Transactions in Operational Research* 26.6, pp. 2324–2336.
196. Tamiz, Mehrdad, Azmi, Rania A, and Jones, Dylan F (2013). "On selecting portfolio of international mutual funds using goal programming with extended factors". In: *European Journal of Operational Research* 226.3, pp. 560–576.
197. Tamiz, Mehrdad, Hasham, R, and Jones, DF (1997). "A comparison between goal programming and regression analysis for portfolio selection". In: *Multiple Criteria Decision Making*. Springer, pp. 421–432.
198. Tamiz, Mehrdad, Hasham, R, Jones, DF, Hesni, B, and Fargher, EK (1996). "A two staged goal programming model for portfolio selection". In: *Multi-Objective Programming and Goal Programming*. Springer, pp. 286–299.
199. Tiryaki, Fatma and Ahlatcioglu, Beyza (2009). "Fuzzy portfolio selection using fuzzy analytic hierarchy process". In: *Information Sciences* 179.1–2, pp. 53–69.
200. Tiryaki, Fatma and Ahlatcioglu, Mehmet (2005). "Fuzzy stock selection using a new fuzzy ranking and weighting algorithm". In: *Applied Mathematics and Computation* 170.1, pp. 144–157.
201. Trenado, Manuel, Romero, Mar100éa, Cuadrado, Mar100éa L, and Romero, Carlos (2014). "Corporate social responsibility in portfolio selection: A "goal games" against nature approach". In: *Computers & Industrial Engineering* 75, pp. 260–265.
202. Vansnick, Jean-Claude (1986). "On the problem of weights in multiple criteria decision making (the noncompensatory approach)". In: *European Journal of Operational Research* 24.2, pp. 288–294.
203. Vermeulen, Erik M, Spronk, Jaap, and Wijst, Nico van der (1993). "A new approach to firm evaluation". In: *Annals of Operations Research* 45.1, pp. 387–403.
204. Vezmelai, A, Lashgari, Z, and Keyghobadi, A (2015). "Portfolio selection using ELECTRE III: Evidence from Tehran stock exchange". In: *Decision Science Letters* 4.2, pp. 227–236.
205. Voogd, Henk (1982). "Multicriteria evaluation with mixed qualitative and quantitative data". In: *Environment and Planning B: Planning and Design* 9.2, pp. 221–236.

206. Voogd, Henk (1983). *Multicriteria evaluation for urban and regional planning*. Vol. 207. Pion London.
207. Weber, Charles A and Current, John R (1993). “A multiobjective approach to vendor selection”. In: *European journal of operational research* 68.2, pp. 173–184.
208. Wei, Hechuan, Niu, Caiyun, Xia, Boyuan, Dou, Yajie, and Hu, Xuejun (2020). “A refined selection method for project portfolio optimization considering project interactions”. In: *Expert Systems with Applications* 142, p. 112952.
209. Whitley, Darrell (1994). “A genetic algorithm tutorial”. In: *Statistics and computing* 4.2, pp. 65–85.
210. Wilhelm, Jochen (1980). “Multiperiod portfolio selection and capital asset pricing”. In: *Multiple Criteria Decision Making Theory and Application*. Springer, pp. 487–510.
211. Wu, Liang-Chuan, Chou, Seng-Cho, Yang, Chau-Chen, and Ong, Chorng- Shyong (2007). “Enhanced index investing based on goal programming”. In: *The Journal of Portfolio Management* 33.3, pp. 49–56.
212. Xidonas, Panos (2010). “An integrated methodology & information system for portfolio management decision support”. Doctoral dissertation. Athens.
213. Xidonas, Panos, Askounis, Dimitrios, and Psarras, John (2009). “Common stock portfolio selection: a multiple criteria decision making methodology and an application to the Athens Stock Exchange”. In: *Operational Research* 9.1, p. 55.
214. Xidonas, Panos, Askounis, Dimitrios, Psarras, John, and Mavrotas, George (2009). “Portfolio engineering using the IPSSIS multiobjective optimisation decision support system”. In: *International journal of decision sciences, risk and management* 1.1–2, pp. 36–53.
215. Xidonas, Panos and Mavrotas, George (2014). “Multiobjective portfolio optimization with non-convex policy constraints: Evidence from the Eurostoxx 50”. In: *The European Journal of Finance* 20.11, pp. 957–977.
216. Xidonas, Panos, Mavrotas, George, Askounis, Dimitrios, and Psarras, John (2008). “Portfolio selection in the presence of multiple criteria”. In: pp.631–648.
217. Xidonas, Panos, Mavrotas, George, Krintas, Theodore, Psarras, John, and Zopounidis, Constantin (2012). *Multicriteria portfolio management*. Springer, New York.
218. Xidonas, Panos, Mavrotas, George, and Psarras, John (2009). “A multicriteria methodology for equity selection using financial analysis”. In: *Computers & operations research* 36.12, pp. 3187–3203.
219. Xidonas, Panos, Mavrotas, George, and Psarras, John (2010). “Portfolio management within the frame of multiobjective mathematical programming: a categorised bibliographic study”. In: *International Journal of Operational Research* 8.1, pp. 21–41.
220. Xidonas, Panos, Mavrotas, George, Zopounidis, Constantin, and Psarras, John (2011). “IPSSIS: An integrated multicriteria decision support system for equity portfolio construction and selection”. In: *European Journal of Operational Research* 210.2, pp. 398–409.
221. Xidonas, Panos and Psarras, John (2009). “Equity portfolio management within the MCDM frame: a literature review”. In: *International Journal of Banking, Accounting and Finance* 1.3, pp. 285–309.
222. Xidonas, Panos, Psarras, John, and Zopounidis, Constantin (2010a). *Modern portfolio theory*. Kleidarithmos, Athens.
223. Xidonas, Panos, Psarras, John, and Zopounidis, Constantin (2010b). *Multicriteria portfolio management: theory and practice*. Kleidarithmos, Athens.
224. Xu, Xiaozhan (2001). “The SIR method: A superiority and inferiority ranking method for multiple criteria decision making”. In: *European journal of operational research* 131.3, pp. 587–602.
225. Yodmun, Satit and Witayakiattilerd, Wichai (2016). “Stock selection into portfolio by fuzzy quantitative analysis and fuzzy multicriteria decision making”. In: *Advances in Operations Research*, pp.1–14.
226. Zopounidis, Constantin (1999). “Multicriteria decision aid in financial management”. In: *European Journal of Operational Research* 119.2, pp. 404–415.

227. Zopounidis, Constantin, Despotis, DK, and Kamaratou, I (1998). “Portfolio Selection Using the ADELAIS Multiobjective Linear Programming System”. In: *Computational Economics* 11.3, pp. 189–204.
228. Zopounidis, Constantin and Doumpos, Michael (2000a). “Building additive utilities for multi-group hierarchical discrimination: The MH DIS method”. In: *Optimization Methods and Software* 14.3, pp. 219–240.
229. Zopounidis, Constantin and Doumpos, Michael (2000b). “INVESTOR: A decision support system based on multiple criteria for portfolio selection and composition”. In: *A-MCD-A Aide Multi Critere a la Decision-Multiple Criteria Decision Aiding*, pp. 371–381.
230. Zopounidis, Constantin and Doumpos, Michael (2002). “Multi-criteria decision aid in financial decision making: methodologies and literature review”. In: *Journal of Multi-Criteria Decision Analysis* 11.4–5, pp. 167–186.