Stochastic mixed-integer programming for financial planning problems using network flow structure

by

Masoud Alimardani

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
Department of Mechanical and Industrial Engineering
University of Toronto

Copyright © 2014 by Masoud Alimardani
Abstract


Masoud Alimardani

Master of Applied Science
Department of Mechanical and Industrial Engineering
University of Toronto
2014

Portfolio design is one of the central topics in finance. The original attempt dates back to the mean-variance model developed for a single period portfolio selection. To have a more realistic approach, multi-period selections were developed in order to manage uncertainties associated with the financial markets. This thesis presents a multi-period financial model proposed on the basis of the network flow structure with many planning advantages. This approach comprises two main steps, dynamic portfolio selection, and dynamic portfolio monitoring and rebalancing throughout the investment horizon. To build a realistic yet practical model that can capture the real characteristics of a portfolio a set of proper constraints is designed including restrictions on the size of the portfolio as well as the number of transactions, and consequently the management costs. The model is solved for two-stage financial planning problems to demonstrate the main advantages as well as characteristics of the presented approach.
Acknowledgments

I would like to express my gratitude to my supervisors Professor Roy H. Kwon and Professor William W. Melek for their influential guidance, inspiration and support throughout my M.A.Sc program. I would also like to thank my family and friends for their sincere encouragements and continuous support.

I also wish to acknowledge the financial supports of the Ontario Ministry of Training, Colleges and Universities.
# Table of Contents

Acknowledgments ........................................................................................................ iii

List of Tables.................................................................................................................. vi

List of Figures ................................................................................................................ vii

Chapter 1  Introduction .................................................................................................. 1

1.1 Research objectives and contributions ................................................................. 3

1.2 Thesis outline .......................................................................................................... 4

Chapter 2  Portfolio optimization ................................................................................ 7

2.1 An overview of basic financial definitions and investment problems ................. 7

2.2 Portfolio optimization and cardinality constraint ............................................... 9

2.3 Stochastic programming for dynamic portfolio optimization ............................. 14

Chapter 3  Network flow structure for portfolio optimization ................................. 21

3.1 Network flow structure and formulation ............................................................. 22

3.2 Network flow and stochastic problems ............................................................... 25

3.3 Application of Network flow models in planning problems ............................... 26

3.4 Stochastic network flow mixed-integer programming for financial planning .... 37

3.5 Numerical results and analysis ............................................................................. 43

Chapter 4  Index tracking ........................................................................................... 52
4.1 Active and passive portfolio management .................................................. 52
4.2 Index tracking models ............................................................................. 56
4.3 Stochastic programming for index tracking ............................................. 63
4.4 A Stochastic network flow mixed-integer programming approach for the index tracking problem ................................................................. 67
4.5 Numerical results and analysis ................................................................. 70

Chapter 5 Conclusion and future research ................................................. 79
5.1 Future research ....................................................................................... 82

Bibliography ............................................................................................... 83
List of Tables

Table 3.1 Return values for three scenarios ................................................................. 43

Table 3.2 Transaction cost for the first stage................................................................. 44

Table 3.3 Transaction cost for the second stage with three scenarios ......................... 44

Table 3.4 Return values generated for three scenarios ................................................. 46

Table 3.5 Transaction cost for the second stage with three scenarios ......................... 47

Table 4.1 Return values for four scenarios ................................................................... 70

Table 4.2 Transaction cost for the second stage with four scenarios ......................... 71
List of Figures

Figure 2.1 Minimum-variance set and the efficient frontier.................................11

Figure 2.2 A schematic of a scenario tree for two time periods .........................16

Figure 2.3 A scenario tree for one time period and ten scenarios ....................17

Figure 2.4 Investment values and their frequencies for different scenarios ..........18

Figure 3.1 A schematic of the network flow structure for: a) undirected and b) directed graphs ..................................................................................................................22

Figure 3.2 A schematic of the network flow structure with additional resource ....24

Figure 3.3 Network flow model for multiperiod financial planning....................27

Figure 3.4 Alternative network flow structure for portfolio management ...........30

Figure 3.5 Two-stage network with four nodes at each stage ...........................42

Figure 3.6 Results for the three scenarios of a portfolio with two securities .......46

Figure 3.7 Results for the three scenarios of a portfolio consisting of three securities..48

Figure 3.8 Number of variables based on the numbers of securities and scenarios: a) s=1,...,10 and b) s=10,20,...,100.................................................................50

Figure 3.9 Number of Constraints based on the numbers of securities and scenarios...51

Figure 3.10 Performance of the model using the commercial solver....................51
Figure 4.1 Results for the three scenarios of a portfolio consisting of three securities .73

Figure 4.2 Tracking error for the portfolio and each scenario ...............................................74

Figure 4.3 Asset allocation: a) with and b) without minimizing the transaction costs 75

Figure 4.4 Number of Variables corresponding to the numbers of securities and scenarios...........................................................................................................................................76

Figure 4.5 Number of Constraints corresponding to the numbers of securities and scenarios...........................................................................................................................................77

Figure 4.6 Running time versus number of securities in the portfolio for n=20 ...........77

Figure 4.7 Gap versus number of securities in the portfolio for n=20 .........................78
Chapter 1

Introduction

Astonishing advances in computers within the last decade provide a wide range of new applications which has been revolutionizing different fields of science and technology. Theoretical developments in finance along with new computational methods and strategies have encouraged many research groups to direct their attention toward further development of the financial management techniques and investment science. Already existing theories and methods along with the new ones have exploited innovative practical means to support and develop new fields and applications. In this fast changing environment, new opportunities can create new complexities and new complexities sometimes can turn into new opportunities. Restricting attention to the financial markets, there have been many national and international market crashes so far with catastrophic consequences like the crash of 1929, or more recently, the crash of 2008-2009. On the other hand, financial markets become considerably a much more complex industry over the last few decades. As a consequence, making investment decisions in selecting securities to form a portfolio becomes a difficult or impossible problem to solve for non-expert individuals. The portfolio selection is the process that starts with observation of the securities and their performances for now and the future. Along with this observation which clearly requires adequate experience, different choices for portfolios are formed. At this stage, beliefs about future performances of different potential choices play the key role for selecting the final portfolio. The nature of this process implies that even many experts in this field can make poor or wrong decisions and many times run into difficulties in making the right choices. Along with this fact, a
brief review of financial market history and circumstances clearly specifies that a portfolio section is a dynamic and complex field which entails multidisciplinary knowledge and expertise. This can be considered the key element in developing more reliable means which can be a proper complement to the experience of portfolio managers gained over years of working in this multifaceted vibrant environment. Methods tailored for this type of fields have been developed to decrease the consequences of wrong or inappropriate decisions. New developments in probability and optimization theories support the decision maker to take the best advantage from available information and data. However, considering all these improvements, all developed quantitative approaches are not still completely reliable due to their shortcomings stemming from volatile financial market environments. This fact should also be taken into account that the human behaviors and their societies cannot fully be prophesied, and therefore, always there are highly unpredicted and uncertain events which can have immense consequences in any aspect of our lives. This is the main reason that on one hand makes the most sophisticated developed means in decision sciences still not fully reliable, and on the other hand has kept this field the subject of intense interest to many researchers all around the world.

Utilization of the practical optimization models in financial decision making as well as risk management were the urge of the complexity and unpredictability governed the financial markets. It was also the fundamental reason for development of the fanatical engineering. Therefore, utilization of the optimization models in finance has only a few decades history like other field of engineering. In financial engineering, although the basic but influential work of Markowitz (1952) had triggered the financial optimization, it had taken a few decades to witness the maturity and advanced utilization of the optimization models in this field. Markowitz was in fact the first one who showed dependency between the risk-adjusted return and the composition of a portfolio.

There is a twofold challenge in developing a robust dynamic model for financial optimization. One side of the main challenge is the definition of a realistic underlying probability distribution that can represent the market behaviour as realistically as possible without overlooking existing uncertainties. The other side is to develop an
optimization model to integrate the stochastic nature of the related market parameters with other characteristics of a portfolio selection and management.

1.1 Research objectives and contributions

The main objective of this thesis is to develop a model for portfolio design and management that can have the capability of considering the main elements of a real world problem including market uncertainties. Outlining the investment goals is the first step for a portfolio design. However, coordinating these goals with circumstances of the financial market is the next step which makes this process highly complicated. Short and long terms investment goals and decisions should be incorporated in this process considering the current and future market environment. The portfolio allocation with all its relative strategic steps is performed based on a set of assumptions of the financial market. These assumptions are dynamically reviewed and modified against new data of the market conditions. This should be done to ensure that the investment portfolio is still reflecting the main goals of the investors, and if not, the right decision for any required modification should be made at the right time. Considering all these facts, to develop a method that can use the most reliable decision making process based on the available data is not normally an easy and straight forward task. Therefore, designing a comprehensive model which can be used for this process usually results in a complex mathematical model with computationally intensive and hard constraints. To overcome some of the difficulties associated with the solution process of these models, researchers and practitioners in this field have made some assumptions to simplify and consequently to solve their developed models. Although, these attempts have shown many advantages toward a better understanding of the portfolio design process, their results were not reliable, and consequently could not be used for real applications. However, as pointed out recent advances in computational means and methodologies have been of great assistance in developing and solving relatively more realistic approaches.

In this thesis, a modelling approach is presented by which it is possible to model financial planning problems in a realistic fashion. To achieve this goal, network flow structure is used to build the proposed approach. This type of structure offers many
advantages that can be used to capture many important features of the planning problems. The developed model encompasses the main attributes of a well-designed investment portfolio such as diversification, dynamic monitoring and rebalancing of the designed portfolio, and implementation of uncertainties associated with the future circumstances of the financial markets or investors’ goals. Most importantly, restriction on the investment costs, considering cardinality constraint and minimizing the transaction costs, is another important contribution of this research. For this purpose, the objective function is restricted by the constraints defined to capture the practical nature of the financial problems. The same framework used for the portfolio design problem is applied to introduce an index tracking modelling approach. Index tracking problem is designed as a passive portfolio management strategy which can assist portfolio managers to compare the performances of their portfolios with a specified benchmark. This is a popular and reliable method for long term investments.

1.2 Thesis outline

This thesis comprises five chapters. The first chapter provides a brief introduction to the financial optimization models as well as the main objectives and motivations of this research. Other subsequent chapters of the thesis are organized as follows:

Chapter 2 - Portfolio theory and optimization

In this chapter, a brief overview of the basic financial definitions is presented along with the main aspects of the portfolio selection and optimization. The mean-variance optimization model as a fundamental concept using quadratic programming for portfolio selection is reviewed. Some of the main models developed based on this concept were also addressed through a brief literature review accompanied by some basic characteristics of the investment portfolios. Another section of this chapter is allocated to stochastic programming for dynamic portfolio optimization. The main techniques used for scenario generation of multistage decision problems are also reviewed in this section.
Chapter 3 - Network flow structure for portfolio optimization

Network flow problems are considered as one of the most popular problems which have been modelled and solved among linear programming problems. This chapter starts with a review on the network flow structure. The main characteristics of the network flow are described and their mathematical formulations are then presented. The main advantages of this type of structures for several planning problems are pointed out as well. It was described how the network flow can be used to capture the uncertainties associated with some real life problems. Financial planning problems are described in more detail using the network flow structure. It is outlined how the special characteristics of this structure can be used to fully encompass the main features of financial planning problems along with their associated uncertainties. For this purpose, uncertain parameters as well as design variables of a financial problem are first introduced and then the problem formulation is discussed. A section is also allocated to briefly describe a novel approach used for the commodity network design. Ultimately, this chapter ends with a network flow stochastic mixed-integer programming model proposed for the financial planning. The main advantages of the proposed approach are listed through solving a two-stage model and analyzing its corresponding numerical results.

Chapter 4 - Index tracking

Chapter 4 describes index tracking problems in which a portfolio manager can evaluate the performance of her or his portfolio by comparing it with a target value as a benchmark. The first section of this chapter is allocated to a literature review. Some of the basic definitions as well as important elements of a decision support system are outlined. It is described that a tracking portfolio with a cardinality constraint as a part of this decision support system is considered NP-hard, and therefore, available index tracking models do not consider all important elements pointed out in this section. In addition, basic models and objective functions for the index tracking problems are addressed and the stochastic programming for dynamic index tracking problems is also briefly discussed in this chapter. Similar to Chapter 3, in this chapter a stochastic network flow mixed-integer programming approach for the index tracking problem is
presented as well. To show different aspects of this proposed modeling approach, a two-stage model is solved and the related results are presented and discussed.

Chapter 5 – Conclusion and future research

Finally in this chapter, a summary of the work described in the preceding chapters is presented. An outline of future research directions based on the work presented in this research is briefly presented in the final section of this chapter.
Chapter 2

Portfolio optimization

At the time of the investment, there is no definite certainty for the return value of a designed portfolio. There are also many different types of risks involved in any investment depending on the nature of the investment as well as the market. However, behind any investment there are at least few design goals. In order to manage the risk and any uncertainty associated with the investments as well as to meet the desired design goals, many mathematical techniques and optimization models have been developed within last few decades. In this chapter, some of the basic however central definitions and mathematical models in financial engineering are addressed.

2.1 An overview of basic financial definitions and investment problems

Interest rate is the price charged for the use of assets and basically is defined based on the percentage of the principal. Consequently, it is a basis for investors to study and consider their investment alternatives. In a financial market, the available items are known as financial instruments. A financial instrument can be defined as a real or virtual legal document which carries monetary value and ones that can be traded in the market are called security. There are many different types of financial instruments and securities that can be traded in the financial market. More details in this regard can be found in the literature [2, 3]. Basically investment problems are more complicated than they seem. The complexity arises from different natures of the securities, investors’
goals, unexpected risk, and uncertainty associated with different elements of any investment portfolio.

One of the main investment problem categories is security pricing. Security pricing is considered as one of the significant achievements of modern financial economics by which it is possible to predict the price of securities through developed mathematical tools. This is the process that can assist investors to make decisions for building a portfolio in a complex market environment. For this purpose, many stochastic principles have been adopted, such as Martingale and Brownian motion [1], to design techniques of predicting future security values. In addition and parallel to this approach, other techniques had been developed by which it is possible to measure the confidence in a stock future price instead of predicting the future price of securities. One of these measures is the beta value of a particular security which is a measure of its associated price risk, a more detailed review regarding this topic is provided in the literature [2]. Another investment problem is hedging. Hedging is a process that is used for reducing the risks associated with the investments and it is considered as one of the most important aspects of the modern financial industry. Another process for reducing the risk associated with a portfolio is diversification. A well-diversified portfolio has a reduced risk exposure throughout the investment horizon. There are many other types of financial problems one can encounter in the field of financial engineering [3].

In terms of pricing, for instance, pricing bonds is very important for the bond buyers to decide whether to buy a specific bond or not. A bond represents an obligation of the company issuing the bond to pay money to the bond holder according to a set of specified rules. Most bonds pay periodic coupon payments as well as their face values. Therefore, pricing a bond is the present value of the cash stream expected to receive during a specific time period or simply is the present value of the bond principal and its coupons. Therefore, the present value of bonds can be calculated as:

\[ P = \sum_{i=1}^{n} \frac{C}{(1 + r)^i} + \frac{F}{(1 + r)^n} \]  

(2.1)
where \( C \) is the periodic coupon payment, \( r \) is the interest rate, \( n \) is the time period for each payment, and \( F \) is the face value of the bond. It is possible after having the present value of a bond to determine its other characteristics such as its yield. The bond yield is determined by solving an equation defined for the present value of bonds considering a constant interest rate. Therefore, the yield of a bond is the interest rate applied to the bond’s stream of payments to be equal to the present value or current price of the bond. However, pricing bonds is much more complicated and interesting when interest rates are not constant and in fact there are uncertain rates of interest. In this field, there are many comprehensive developed models to price bonds considering the value and risk of the firm’s assets by which bonds are issued. It should be considered that there are many uncertainties associated with the future risk and value of the firm’s assets. One can consult the literature for more description and details on the developed models and related issues of pricing bonds [4, 5]. The variation in yields for different bonds can partially be justified through various quality ratings of bonds. Therefore, this fact implies that two bonds with the same promised income streams have different values if for instance one belongs to the strong AAA rated bonds and the other to the B quality rated bonds. This rating indicates the default risks associated with the interest rate related to a bond and consequently its income stream [3].

2.2 Portfolio optimization and cardinality constraint

One of the most influential works in financial optimization which has had a definite impact on the field of financial optimization is the mean-variance optimization model (MVO). This model was developed by Harry Markowitz [6] in 1952 which was also the main foundation for modern portfolio theory. In this model, the concept of minimizing variance to design a portfolio with minimum risk and its correlation with diversification are discussed (i.e., portfolio return versus risk). More specifically, MV analysis investigates the compromises between portfolio returns (i.e., portfolio reward) and their associated risks. The reward is determined by the expected return of the portfolio, and the risk is considered by the portfolio variance. MV analysis was developed for being a positive as well as a normative tool. For instance, for a positive tool, the Capital Asset
Pricing Model (CAPM) is one of the MV analysis results. It was also reported that the MV portfolio theory was used to develop a mental accounting framework by which it is possible to create a portfolio to consider the expected return and risk along with the investors’ goals. On the other hand, as a normative tool, it can establish a basis for the investors to behave in the market. The main contribution of the normative MV analysis is constructing an efficient portfolio with higher expected returns than all other possible options while having the same level of risk \[2, 7\].

To formulate the MVO problem, let consider that there are \(n\) assets with the expected rate of return of \(\mu_1, \mu_2, \ldots, \mu_n\) and corresponding covariances of \(\sigma_{ij}\) for \(i\) and \(j = 1, 2, \ldots, n\). A portfolio is selected by a collection of assets which their weights \(x_i\) for \(i = 1, 2, \ldots, n\) sum to 1. The MVO problem can be formulated as:

\[
\begin{align*}
\min & \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \\
\text{s.t.} & \quad \sum_{i=1}^{n} \mu_i x_i = R \\
& \quad \sum_{i=1}^{n} x_i = 1
\end{align*}
\] (2.2)

where \(R\) is the target expected return of the portfolio. The first constraint requires the average portfolio return to be equal to the target expected return. The MVO model stated above is simply a quadratic model with linear constraints. In some versions of the MVO model, the first constraint requires the average portfolio return to be equal to or larger than \(R\). Here \(R\) is the lower bound that would be active at an optimal solution. However, the model can be formulated as presented here without loss of generality. In the above MVO model since the negative weights can be allocated to the assets, the short selling is possible. If the short selling is not allowed, then another constraint should be added to the constraint set to impose restrictions on the variables. This constraint can be defined as:
In the MVO model, it is possible to generate different portfolios by changing the target expected return, $R$. The left bound of the feasible set is known as the minimum-variance set and has a bullet shape. In addition, there is a point on the boundary of the feasible set that has the minimum variance and is called the minimum-variance point. By solving the MVO model for different target expected return values, it is possible to plot the minimum-variance set in the risk-return plane. The upper portion of the minimum-variance set is known as the efficient frontier that shows the outermost line of the optimal portfolios. Figure 2.1 schematically depicts the minimum-variance set, the efficient frontier, and the minimum-variance point. Although most of the optimization models developed recently encompass more sophisticated details including risk management in the financial optimization process than MVO model, MVO analysis is still one of the most influential tools which are widely in use today.

![Figure 2.1 Minimum-variance set and the efficient frontier.](image)

As pointed out, the CAPM was developed within MV analysis framework by which it is possible to realize the correct price of a risky asset. This model was developed by Sharpe, Lintner, and Mossin [8, 9, 10]. The CAPM is mainly suggesting that the securities should be priced at the equilibrium in a way that their expected returns will be a linear combination of the market return and the risk-free return.
As pointed out diversification is considered an advantage in the MVO and CAPM models as investing in a large array of assets results in reducing associated risks. Therefore, to some extent, all investors diversify their investment relative to their degree of risk-aversion. However, this advantage might generate many difficulties associated with the portfolio management and maintenance such as high cost of rebalancing (i.e., transaction cost) and tracking the performance of all assets in the portfolio. As a result, having a limitation on the number of securities included in a portfolio is an important aspect of building a proper portfolio [11, 12, 13]. In this regard, for instance, Coleman et al. [14] proposed a method to minimize the tracking error with a specific total number of assets. The corresponding constraint, to limit the number of securities which is also called cardinality constraint, can be expressed as:

$$\sum_{i=1}^{n} H(x_i) \leq K \quad (2.6)$$

where $H(x_i) = 1$ for securities used in the portfolio (i.e., $x_i \neq 0$), otherwise $H(x_i) = 0$, and $K$ is an upper bound on the number of securities included in the portfolio. If the objective function of the model is defined to measure the tracking error (i.e., trying to replicate a particular behavior), the cardinality constraint can be considered as the upper bond imposing on the administration costs. It should be noted that having cardinality constraints requires the presence of binary variables in the optimization problem formulation. Hence, the optimization problem changes from a quadratic program (QP) to a quadratic mixed integer program. Solving optimization model including the cardinality constraint is NP-hard, and consequently solution algorithms designed for this type of problems are mainly heuristic in nature. For the developed solution strategy a detailed description is provided by Coleman et al. [14], however more details regarding the relaxation of the cardinality constraint and solution algorithm of this type of problems are addressed in the next chapters. In another research program, Jobst et al. [15] investigated the effect of considering cardinality constraints in the MV model for portfolio selection. It was pointed out that the cardinality constraints also imply the buy-in threshold. It defines a minimum level for an asset that cannot be
purchased below that level. This is the constraint by which the unrealistic small trades are eliminated from a portfolio. Some heuristic algorithms and newly proposed approaches were also developed for incorporating the cardinality constraint in the portfolio optimization model [16, 17, 18, 19]. Some investigations were also conducted to develop models with more realistic and comprehensive constraints. For instance, DeMiguel et al. [20] studied the condition under which MV analysis for optimal portfolio model can be considered to perform well with the presence of the estimated risk. For this purpose, they evaluate the out of sample performance of the MV sample based optimization model. Alexander et al. [21] also examined the effect of including either a VaR or a CVaR into the constraint of the MV model in which the security returns were considered to have a discrete distribution with finitely many jump points. It was suggested that a CVaR constraint is more efficient than VaR constraint when considered in the MV model to restrain large losses. Furthermore, to be taken into account the higher moments of return distributions rather than using just the first two ones, Liu et al. [22] investigated an MV skewness model for portfolio selection with transaction cost. This is done as performed in many other studies like the one conducted by Chunhachinda et al. [23]. It was claimed that the higher moments should be considered unless they are not related to the investors’ decision making process, or the security returns are normally distributed and utility function is quadratic as well. In their work, the difficulty resulted from non-smoothness was overcome by transforming the problem into a linear programming problem. In some other investigations regarding portfolio selection within MV analysis framework, Lagrangian relaxation algorithm was applied to solve cardinality constrained quadratic programming problems [24, 25].

Although the seminal mean-variance analysis [6] and other influential researches conducted in the 60s [8, 9, 10] are still playing important roles among current available practical tools for portfolio optimization, they have some important shortcomings. For instance, mean-variance analysis is only for a single period portfolio selection or it does not consider the current and future circumstances of the market [2, 3]. Therefore, due to all these shortcomings and growing complexities of the financial markets like other areas, many researchers and practitioners have been motivated to develop more versatile
and reliable methods. Hence, the movement for the portfolio management that has been started from the classical static mean-variance analysis has been directed towards scenario-based approaches, and consequently multi-period dynamic portfolio optimization strategies, which is the main focus of this research.

Since in multi-period planning problems there are different decision stages in which successively uncertain information is realized, these types of problems have stochastic nature. For this purpose and to reflect this fact in the structure of a dynamic planning problem, a set of scenarios can be generated. Before moving to the next chapter and describing the proposed dynamic optimization strategy, the following section briefly reviews the main techniques that can be applied in financial planning for generating a set of scenarios.

2.3 Stochastic programming for dynamic portfolio optimization

In contrast to a deterministic process in which all essential information is available when the required decisions are made, a stochastic process is used to consider the evolution of random values over time. Therefore, the latter type of processes deal with the situations in which decision making is performed under uncertainty. Explicitly in an optimization problem, it is a means of introducing stochastic variability within parameters of the problem, and consequently finding an optimal decision in a problem with a set of uncertain data. An important category of stochastic programs is those in which some decisions can be made after uncertainty is unveiled which are called recourse programs. As pointed out, random variables represent the uncertainty involved in some data with available probabilistic description. In the simplest form, two-stage stochastic model, the decision sets can be grouped into the first and second stage decisions. In the former stage, some required decisions should be made before random events take place (i.e., first-stage decision). A number of decisions should then be made after the realization of the random variables. This should be done in order to adapt the solution obtained by the first-stage decisions to the random variables which have been taken place. In the stochastic models, it is of great importance to represent uncertainties properly for the computation throughout the solution process, as shown in the following basic model [25]:

\[\text{Basic model} \]
\[
\min c^T x + E_\Omega[\min d(\omega)^T y(\omega)]
\]  
(2.7)

\[
s.t. \quad Ax = b
\]  
(2.8)

\[
P(\omega)x + By(\omega) = q(\omega)
\]  
(2.9)

\[
x \geq 0, \quad y(\omega) \geq 0.
\]  
(2.10)

where \(x\) is a vector of length \(n_1\) and the first stage decisions, \(c\) and \(b\) are vectors of length \(n_1\) and \(m_1\), and \(A\) is a \(m_1 \times n_1\) matrix. In addition, for a random event \(\omega \in \Omega\) in the second stage, vector \(d(\omega)\) of length \(n_2\), vector \(q(\omega)\) of length \(m_2\), and \(P(\omega)\) a \(m_2 \times n_2\) matrix become known. The objective function consists of a deterministic term and \(E_\Omega[\min d(\omega)^T y(\omega)]\) which is the expectation of the second stage objective for \(\omega \in \Omega\).

It should be noted that time periods and stages in a stochastic process are different, and one may not exactly coincide with the other. However, in portfolio management, for instance, different stages can simply be determined according to the designated time periods in which the performance of a portfolio is evaluated (daily, weekly, etc.). In the stochastic programming, calculation of the recourse function, or in other words, the probability that the constraints of a problem can be satisfied is extremely complicated and in several cases impossible. In typical differentiable numerical problems, it is possible to apply some form of approximation to overcome the involved complexity. However, stochastic problems are not mainly differentiable but some useful convexity properties can assist to alleviate this difficulty. Furthermore, it is possible to approximate a problem through its discretization of the related probability distributions of its random variables, and consequently generating a set of scenarios. Each scenario, which represents associated uncertainties, basically shows a realization of the random events. For multistage stochastic process, sampling methods are very common as the number of possible scenarios increases exponentially in this type of problems. During scenario generation, consideration should be given to ensure that this method provides a solution with similar characteristics to the main problem.
Different techniques for scenario generation are developed and addressed in the literature [2, 27, 28, 29, 30]. Some of these techniques were developed for implementing a stochastic model for asset allocation in which scenario generation is considered as the discrete outcomes of the random events [2, 27, 28]. However, these methods can also be applied for different types of decision making models under uncertainty. As mentioned, a scenario is a representation of all random variables which are realized for all time periods. A scenario tree for two time periods is schematically shown in Figure 2.1. In a tree, a node is a state of a problem in a specific time in which any particular decision can be made, and each arc is considered for realization of the uncertain variables. The root node which can be observed from available deterministic data represents current state of the variables. Therefore, a path through the tree is a scenario as shown in Figure 2.2 for the scenario number 4. The whole tree depicts all possible outcomes of the random variables which comprises both optimistic as well as pessimistic potential outcomes of the process under investigation.

![Scenario Tree](image)

**Figure 2.2** A schematic of a scenario tree for two time periods

As pointed out, there are different methods to generate scenarios considering the data available before the starting point of the investment, methods such as Monte Carlo (i.e., generating random numbers from the assumed distribution) or bootstrapping (i.e., generating randomly from historic scenarios). For instance, using Monte Carlo simulation model, returns of securities for the first stage are calculated based on the statistics of the available data. By adding the results obtained for the first stage to the
available data, the Monte Carlo simulation model can be recalibrated for the second stage calculations. This process should be repeated until the whole investment time horizon is covered. Using this method, for instance, Figure 2.3 shows the results of 10 scenarios generated from normal distribution of S&P 500 index market over the last five years (2008-2013) for one time period investment of 1000 dollars. Figure 2.4 also shows the frequency of random distribution of the investment after one period.

There are many advantages of using historic data to generate scenarios, for instance for the risk evaluation. The historic data are plausible as they have already occurred, and also are easy to generate. It should be noted that the traditional Monte Carlo simulation is very time consuming and also depends on a predefined distribution. There are different methods to reduce the number of scenarios without compromising the accuracy intended to have for the risk estimation [2, 27].

Figure 2.3  A scenario tree for one time period and ten scenarios
As pointed out, statistical properties of the random variables are the key point for the scenario generation. Since in many cases the random events are not discrete and their underlying distributions are not known as well, a well-designed procedure is required for the scenario generation. In the remaining part of this section, a description of a method developed by Hoyland et al. [28] for scenario tree generation is briefly presented. In this method, they developed an optimization approach for generating scenario tree. For this purpose and to describe their proposed method, all statistical properties are denoted by the set $S$, and $S_Y$ is the value of a particular statistical property $k$ in the set $S$. In addition, $I$ is the number of random variables, $T$ is the number of stages. Corresponding to an outcome vector $x$, $p$ is the probability vector with dimension: $n_1 + (n_1 \times n_2) + \cdots + (n_1 \times n_2 \times \cdots \times n_T)$ where $n_t$ is the number of outcome in stage $t$. $f_k(x, p)$ is the mathematical expression for statistical property $k$. The main goal here is to match a set of approximated statistical properties to a set of specified statistical properties by constructing proper $x$ and $p$. This goal can be achieved by minimizing the distance between distributions of two aforementioned sets of statistical properties. This minimization model is subjected to a set of constraints in which all probabilities should sum up to one and also be nonnegative. The distance is measured by the square norm as follows:

![Figure 2.4 Investment values and their frequencies for different scenarios](image.png)
\[
\min_{x,p} \sum_{k \in S} w_k \cdot (f_k(x, p) - S_v)^2 
\]
\[\text{s.t. } \sum p.A = 1 \] (2.12)
\[p \geq 0.\] (2.13)

where \( A \) is matrix with the columns by which a conditional distribution of a statistical property in the scenario tree is represented. This minimization problem may not converge toward a perfect match due to the high inconsistency throughout specifications or nonconvexities of the optimization problem. Therefore, one can simply adjust the weight, \( w_k \), of each statistical distribution according to their importance considering the specifications of the problem. It should be pointed out that the statistical properties are including mean, variance/covariance, third central moment (i.e., skewness), and fourth central moment (i.e., kurtosis). Although due to the nonconvexity the optimum values achieved through the minimization problem might be local, they are still satisfactory as the distribution properties are close to the desired specifications.

For the asset allocation problems, the scenario tree represents the uncertain returns in different time periods for different asset classes, which is also the fundamental step in dynamic portfolio management. As mentioned in the preceding sections, the main purpose of a dynamic portfolio management is to achieve some long range goals by which decision consequences amortize over several time periods. The technique described here is also capable of capturing the phenomenon called volatility clumping which indicates a period with high volatility that can be followed with a similar situation. For the uncertain returns in financial market, the historical data vectors are known for different assets. More details in this regard as well as numerical examples for the presented technique are provided in the literature [2, 27, 28].

Particularly for the financial portfolio problems, three different approaches were developed by Gulpinar et al. [29] for scenario tree generation. Their proposed techniques are developed considering randomly generated scenarios and different variants of the
moment matching methods. For this purpose, a portfolio with \( n \) risky assets was considered. It was intended to restructure the portfolio at discrete times throughout the investment horizon \( T \). For this purpose, past available information along with the probabilistic specifications of the future uncertainty of the random variables were used. For more details regarding the simulation and optimization approaches, one can refer to the literature [28, 29]. In some financial planning applications, scenario tree should be arbitrage-free. Arbitrage is an opportunity in which a positive cash flow is generated during a specific time span without requiring any outflow of fund. In generation of a scenario tree, it is possible to impose some constraints to have arbitrage-free price/return scenarios. This can be done while meeting all desired statistical specifications intended to match the first four central moments and co-moments. Klaassen [30] studied two types of arbitrage opportunities, originally introduced by Ingersoll [31], which can also be detected in the scenario tree for asset prices. They also proposed appropriate constraints that should be added to the nonlinear problem defined by Hoyland et al. [28].
Chapter 3

Network flow structure for portfolio optimization

Network flow models have been utilized to solve and address different aspects of several network planning problems in different application areas such as transportation, cash management, communication systems, and logistics [32, 33, 34]. Network flow problems can be considered as the most referred and popular problems modeled and solved among different linear programming problems. Therefore, they are a particular case of linear programming and any solution strategy developed for this group of problems can directly or indirectly be applied to them. In addition, the developed solution strategies for linear programming applied to this type of problems can noticeably be simplified due to the special structure of the network flow problems. Most solution strategies developed for the network flow problems fall into the three main groups: primal, dual ascent, and approximated ascent methods [35]. A common concept of the network flow models is to distribute a single or a set of commodities from plants (or origins) to destination points (or consumers). Therefore, the main challenge in this type of problems is first to design a network consisting of a set of nodes and directed arcs, the links with the proper features such as capacity restrictions which connect the nodes, and then to minimize the distribution cost accordingly.
3.1 Network flow structure and formulation

Network flow problems as pointed out is a single or a set of resources distributed from a single or multiple starting points to the several destination points. Therefore, based on this concept, it is possible to define and represent the network flow problems on graphs. The graph representing a problem can be undirected or directed. Figure 3.1 exhibits the main structure of an undirected and a directed network flow graph.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{network_graph.png}
\caption{A schematic of the network flow structure for: a) undirected and b) directed graphs}
\end{figure}

For an undirected graph with a node set $N$ and a set arc $U$, which are also called edges, arcs are undirected and unordered. In addition, both nodes through which an edge is connected to the network are called the endpoints. On the other hand, for a directed graph $G = (N,A)$ with a node set $N$ and arcs set $A$, arcs are directed. The order of the nodes connected with an arc shows the direction of the corresponding arc. For instance, an arc between two nodes $i$ and $j$, can have order of $(i,j)$ or $(j,i)$ depending on its direction, outgoing from node $i$ and incoming to node $j$ for the former case and
outgoing from node \( j \) and incoming to node \( i \) for the latter ordering case. It is simply possible to build an undirected graph from a directed graph by removing the direction of the arcs as well as repeated arcs, the same as it was done for the graph shown in Figure 3.1. Contrary to the undirected graph, for a directed graph flow is only possible from an outgoing node to an incoming node. For both directed and undirected graphs self-arcs (i.e., one node is the outgoing and incoming node for an arc) is not permitted. In a graph, a walk is defined as a collection of arcs connected through distinct consecutive nodes, \( n_1, \ldots, n_k \), and a path is a walk without any nodes repeated. In this context, a cycle is formed from a walk if the first node and the last node of the walk become identical, \( n_1 = n_k \).

Another concept in the network flow which is erected on the aforementioned definitions is the tree. A tree is referred to a connected undirected graph without any cycle in its structure. A leaf of a tree is then a node with only one arc connected to it. It can also be said that a leaf is a node with degree of 1, as the degree of node indicates the number of the arcs connected to a node in an undirected graph. Spanning tree is a tree formed on an undirected graph \( G = (N, U) \) when a subset of \( U \), let call it \( U_1 \), makes a tree such that \( T = (N, U_1) \). At this stage and relying on the definitions presented in this section, it is possible to formulate a network flow problem.

A network flow is represented by a directed graph \( G = (N, A) \) and might have external resources or supplies as is schematically shown in Figure 3.2. Each arch \((i, j)\) has an upper bound or capacity, and also the cost per unit of flow. Considering a network flow problem, the following condition can be imposed on the variables of the problem, \( x_{ij} \), which is the amount of the resources or commodities flowed through an arc \((i, j)\).

\[
b_i + \sum_{j \in I} x_{ji} = \sum_{j \in E} x_{ij}, \quad \forall \ i \in N \tag{3.1}
\]

\[
0 \leq x_{ij} \leq u_{ij} \quad \forall \ (i, j) \in A \tag{3.2}
\]
where $b_i$ is the external resource or commodity, $u_{ij}$ is the capacity of each arc $(i,j)$. In addition, $I(i)$ and $E(i)$ are the sets of start and end nodes of arcs, respectively. The first equation is the flow conservation law. The flow conservation law indicates that the total amount of flow entering into a node should exactly be equal to the amount of flow exiting that node. The second equation simply states the non-negativity and upper bound constraints imposed on the flow of each arc. If the conservation equation is considered through all network, the result is $\sum_{i \in N} b_i = 0$, which express the fact that all flow entering into the network through external resources is equal the flow leaving the network and entering to the environment. This condition always should be held, otherwise, the network flow problem would be an infeasible problem.

![Figure 3.2 A schematic of the network flow structure with additional resource](image)

Now, it is possible to define an objective function of a network problem for the minimization of the total linear cost of the flow throughout the network as follows:

$$\sum_{(i,j) \in A} c_{ij} x_{ij}$$  \hspace{1cm} (3.3)

where $c_{ij}$ is the cost per unit of flow for arc $(i,j)$. It should be noted that if the flow in the arcs is not bounded, there is only non-negativity constraints as well as other constraints which are equality constraints, and therefore, the corresponding linear problem is in standard form. In this case, it is possible to apply any related algorithm developed to solve this type of problems such as the network simplex algorithm. Network problems with no bound on the capacity of their arcs are called uncapacitated problems, otherwise they are called capacitated problem, problems with upper bound constraints on the flow capacity across their arcs.
Based on the information provided in this section, now one can formulate different problems or special cases such as maximum flow problem and transportation problem based on the network flow problem characteristics. More detail in this regard and the algorithms developed for solving network flow problems are provided in the literature [35].

3.2 Network flow and stochastic problems

A large number of problems were simplified to deterministic models in which all required information was assumed to be available. However, it should be noted that the results of the deterministic problems in which the uncertainties associated with the random variables are ignored differ significantly from the optimum solutions of its corresponding real world stochastic problems [36]. The main difficulty involved in solving stochastic problems has been the size of the problems and the complexity involved in the solution process of this type of problems. As mentioned, the size of a stochastic problem is a function of the number of decision stages as well as the number of realizations of the random variables associated with the problem. This was the main reason for developing the solution algorithm primarily for the problem with special structures rather than a comprehensive algorithm encompasses real world problems. This fact should also be taken under consideration that the network structure of a stochastic problem is destroyed when it is transformed to its explicit equivalent deterministic form [37]. As pointed out in the preceding section, stochastic problem decomposition through scenario generation converts the main stochastic problem into a manageable number of subproblems. In addition, it is very important to be able to solve large size problems. Progressive hedging (PH) algorithm is one of the efficient solution strategies developed by Rockafellar and Wets [38] based on the principle of scenario aggregation. Using this method, it is possible both to manage subproblems, which are generated through scenarios built based on a relatively large scale stochastic problem, and also preserve special structure of the main problem across generated subproblems. Mulvey et al. [39] used this method to solve a two-stage stochastic network flow problem; however, it can simply apply for multistage problems as well. More detail
regarding the PH algorithm and its related theories as well as its structural properties is available in the literature [38, 40]. In PH algorithm and for each scenario, the nonanticipativity constraints (i.e., constraints which impose decisions that do not depend on the future realization of the random event at a specific stage) are relaxed by placing them in the objective function using multiplier elements and penalty terms. These constraints are then progressively enforced through iterative method until to reach to a consensus among all scenario subproblems. In this way, the main stochastic problem is decomposed to a collection of the smaller subproblems defined based on the different scenarios while maintaining the network structure. In this method, the penalty parameter should iteratively be adjusted, if required, until the termination criteria are met. It was pointed out that the dynamic adjustment for the penalty parameter can have significant computational benefits. It was also described that PH algorithm does not guarantee the primal feasibility within intermediate steps resulted from the relaxation of nonanticipativity constraints. More details in this regards as well as implementing issues are provided in the literature [37, 38, 39, 40]. Mulvey et al. [41, 42] applied PH algorithm for the solution of the financial planning problems defined based on the stochastic network programming protocol.

3.3 Application of Network flow models in planning problems

In a network flow financial planning problem, each node associated with an asset at each time step, and each arc of the network denotes a transaction decision. Transaction decisions normally have a multiplier that can scale the flow passing through each arc. Therefore, rate of return, exchange rate, and any other similar scaling parameters can be represented by an arc multiplier across the network. Supply and demand arcs in the network show external deposits as well as withdrawals. Furthermore, the flow conservation at each node also indicates a balance among the external sources as well as users (i.e., external deposits and withdrawals) and the internal flows throughout the network within any specific time horizon. As pointed out, in the financial planning any realization of the uncertainties associated with the decision variables is represented by a distinct scenario as well. Mulvey et al. [41, 42] applied network flow structure for the
financial planning problems. To briefly describe their proposed approach, each scenario is denoted by $s$ and correspondingly $S$ represents a discrete set of scenarios. Let $p_s$ be the probability associated with the occurrence of scenario $s \in S$, and therefore:

$$\sum_{s \in S} p_s = 1 \quad (3.4)$$

where $p_s > 0$. External sources are denoted by $d_t^i(s)$ for $s \in S$, time period $t = 1, 2, ..., T$ and node $i = 1, 2, ..., n$ that $d_t^i(s) > 0$ and $d_t^i(s) < 0$ indicate external inflow and outflow, respectively. Before proceeding further, for better understanding of the problem, Figure 3.3 schematically shows the network flow of the problem.

![Figure 3.3](image)

**Figure 3.3** Network flow model for multiperiod financial planning

For each asset $i \in N$ and $N_i = N \setminus \{i\}$, and also $x_{ij}^{t_2 t_1}(s)$ represents a transaction between two nodes $i, j$ at the period $t'$ that $t < t' < T$. $t'$ shows transaction for the future which its decision is made at the time $t$, if $t = t'$ the intended transaction is made at the same period of decision making. $e_{ij}^{t_1 t_2}(s)$ is the multiplier on the arc between nodes ($i$ and $j$), which shows the transaction or forward exchange rates. On the network, any particular asset holding can be considered in interest bearing format of
assets that can be shown by \( y_{it'}(s) \), which is the asset \( i \) at period \( t \) with maturity date \( t' \), and consequently \( r_{it'}(s) \) denotes the corresponding return rate. It was mentioned that principal investment and its accumulated interest are paid at maturity. In the proposed model, \( u_{it'}(s) \) is the amount of fund that should be transferred from asset \( i \) at period \( t' \) to \( t \) in order to consider borrowing or short sales. Therefore, the available fund at time \( t \) can be calculated as:

\[
\text{Available fund at time period } t = e_{it'}(s)u_{it'}(s)
\]

where \( e_{it'}(s) = \frac{1}{(1 + \theta_{it'}(s))^{(t-t')}} \) and \( e_{it'}(s) \leq 1 \), and also \( \theta_{it'}(s) \) is the interest rate. The loans that matures after the end of the panning horizon (i.e., \( T < t' \)) is also allowed in the developed model. By considering \( w(s) \) as the final total wealth, the vector of the decision variables defined in the problem for scenario \( s \in S \) is \( q(s) = \{x(s), y(s), u(s), w(s)\} \). The general multistage and multiscenario discrete time financial planning model with a general objective function can be formulated as follows:

\[
\max \sum_{s \in S} p_s F(q(s), s) \tag{3.5}
\]

\[
s.t. \sum_{s \in S} x_{ij}^{o0}(s) - \sum_{s \in S} c_{ij}^{o0}(s)x_{ij}^{o0}(s) + \sum_{t' > 0} y_{it'}^{o0}(s) - \sum_{t' = 1} e_{it'}(s)u_{it'}^{o0}(s)
\]

\[
- \sum_{t' > T} (e_{it'}^{o0}(s))^{T-t}u_{it'}^{o0}(s) = d_i^0(s), \quad \forall i \in N, \forall s \in S \tag{3.6}
\]

\[
\sum_{j \in N_i} \sum_{t=0}^{T} \sum_{t' > T} (r_{it'}(s))^{T-t}y_{it'}(s) + \sum_{j \in N_i} \sum_{t=0}^{T} \sum_{t' > T} u_{it'}(s) + w^s_f = 0, \quad \forall s \in S, \tag{3.7}
\]

\[
q(s) \in Q, \quad \forall i \in N, \forall s \in S
\]
\[
\sum_{t=0}^{t} \sum_{j \in N_t} x_{ij}^{tt'}(s) - \sum_{t'=0}^{t} \sum_{j \in N_t} c_{ij}^{tt'}(s)x_{ij}^{tt'}(s) + \sum_{t'=t}^{t} y_{i}^{tt'}(s) - \sum_{t'=0}^{t} r_{i}^{tt'}(s)y_{i}^{tt'}(s) \\
+ \sum_{t'=0}^{t} u_{t}^{tt'}(s) - \sum_{t'=t+1}^{T} e_{i}^{tt'}(s)u_{i}^{tt'}(s) - \sum_{t'=t+1}^{T} (e_{i}^{tt'}(s))^{T-t}u_{i}^{tt'}(s) = d_{i}^{t}(s),
\]

\(\forall i \in N, \forall s \in S, t = 1, \ldots, T\)

As described in the preceding section for the scenario tree, each scenario is defined through a path from root node to a specific leaf node. Each node placed on the path represents a realization of the decision variables at a particular stage and this realization is the same for all different scenarios passing through that node which is denoted by \(m_{t}\). These paths belong to a scenario subset \(L(m_{t}) \in S\) which is those tree paths with some common nodes between the root node and their leaf nodes. In addition, set \(M(t)\) represents the number of disjoint subset at a specific stage which is the number of the tree nodes at that stage. Therefore, by defining the following artificial variables for each non-leaf node \(m_{t} \in M(t)\) for \(t = 0, 1, \ldots, T - 1:\)

\[
X(m_{t}) = \{X_{ij}^{tt'}(m_{t}): \forall i \in N, \forall j \in N_{j}, t \leq t' \leq T\} \tag{3.9}
\]

\[
Y(m_{t}) = \{Y_{i}^{tt'}(m_{t}): \forall i \in N, t \leq t'\} \tag{3.10}
\]

\[
U(m_{t}) = \{U_{i}^{tt'}(m_{t}): \forall i \in N, t \leq t'\} \tag{3.11}
\]

Consequently, the following nonanticipativity constraints can be defined and added to the constraints of the stochastic multistage financial planning model.

\[
x_{ij}^{tt'}(s) - X_{ij}^{tt'}(m_{t}) = 0, \ \forall i \in N, \forall j \in N_{j}, \forall s \in L(m_{t}), \forall l \in M(t), \tag{3.12}
\]

\[
t = 1, \ldots, T, t \leq t' \leq T
\]

\[
y_{i}^{tt'}(s) - Y_{i}^{tt'}(m_{t}) = 0, \ \forall i \in N, \forall s \in L(m_{t}), \forall l \in M(t), \tag{3.13}
\]

\[
t = 1, \ldots, T, t \leq t'
\]

29
\[ u_{i}^{t'}(s) - U_{i}^{t'}(m_t) = 0, \quad \forall i \in N, \forall s \in L(m_t), \forall l \in M(t), \]

\[ t = 1, \ldots, T, t \leq t' \]

The developed model was solved using PH algorithm. For portfolio management, network flow model can be formulated considering a cash node in the structure of the network, and therefore, eliminate direct transactions between each pair of nodes. Consequently, all transactions across the network are conducted through the cash node. This network is schematically shown in Figure 3.4.

![Figure 3.4 Alternative network flow structure for portfolio management](image)

Implementation issues and penalty parameter settings were investigated as well. However, the model does not include any cardinality constraint. More details in this regard and also the modelling approach are provided in the corresponding literature [41, 42].

In one of the recent studies concerning network flow problems, conducted by Crainic et al. [43], a fixed charge capacitated multicommodity network design (CMND) with stochastic demands were mathematically modelled and solved. PH algorithm and a metaheuristic approach were implemented as a part of their proposed two alternative solution strategies. The solution process comprises two steps, designing the network or
making the design decisions, and then using the designed network for realization of the random events. Therefore, the objective function in this two steps problem formulation is to optimize the fixed cost of the design decisions made through the first step as well as the cost of the recourse actions occurring within the second step. Consequently, to design the network $G = (N, A)$, the problem can be formulated as follows:

$$\min \sum_{(i,j) \in A} f_{ij} y_{ij} + E_d [Q(y, d(\omega))]$$

where, $f_{ij} \forall (i,j) \in A$ is the fixed cost for each arc opened, $y_{ij} \forall (i,j) \in A$ is equal to 1 if the arc is open and zero if it is closed, $E_d [Q(y, d(\omega))]$ is the distribution cost for a given design while $d(\omega)$ denotes the recourse cost. For a given random event $\omega \in \Omega$ and a commodity $k \in K$, $d^k_t(\omega)$ can be considered as:

$$d^k_t(\omega) = \begin{cases} v^k(\omega) & \text{if } i = o(k) \\ -v^k(\omega) & \text{if } i = s(k) \\ 0 & \text{otherwise.} \end{cases}$$

The main contribution of this work was first to propose a metaheuristic framework based on the progressive hedging algorithm by which it is possible to take advantage of methods developed for solving deterministic CMND problems. Furthermore, different strategies were also developed for penalizing non-consensus among scenario subproblems in order to approximate the global design. Performances of their developed strategies for penalizing were then compared and analyzed through different experimental results. It was pointed out that in the proposed approach the first stage design decision cannot be changed during the second stage in which the distribution of the commodities are made. CMND problems have mainly been investigated using deterministic models in which all required information is available when the design decisions are made. However, the optimization problems are formulated possessing different decision stages at which uncertain information is known. Therefore, CMND problems in real-life have mainly stochastic nature which was considered in the structure of this work.
Once the first stage design decisions are set, the term $E_d[Q(y,d(\omega))]$ related to the commodity distribution in the objective function can be represented as follows:

$$Q(y,d(\omega)) = \min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k$$  \hspace{1cm} (3.18)

subject to:

$$\sum_{j \in N^+(i)} x_{ij}^k - \sum_{j \in N^-(i)} x_{ji}^k = d_i^k(\omega), \quad \forall i \in N, \forall k \in K, \quad (3.19)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i,j) \in A \quad (3.20)$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, \forall k \in K \quad (3.21)$$

where $x_{ij}^k \forall (i,j) \in A$ and $k \in K$ is the amount of commodity flows through an arc, and $c_{ij}^k$ is the corresponding cost. First constraint is the network flow conservation constraints. The second and the third constraints apply the flow capacity of each arc of the network, and the non-negativity of the variables, respectively. By replacing $Q(y,d(\omega))$ defined through the aforementioned minimization problem into the original model, the deterministic model of the CMND problem can be written as:

$$\min \sum_{s \in S} p_s \left( \sum_{(i,j) \in A} f_{ij} y_{ij}^s + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^{ks} \right)$$ \hspace{1cm} (3.22)

subject to:

$$\sum_{j \in N^+(i)} x_{ij}^{ks} - \sum_{j \in N^-(i)} x_{ji}^{ks} = d_i^{ks}, \quad \forall i \in N, \forall k \in K, \forall s \in S \quad (3.23)$$

$$\sum_{k \in K} x_{ij}^{ks} \leq u_{ij} y_{ij}^s, \quad \forall (i,j) \in A, \forall s \in S \quad (3.24)$$

$$y_{ij}^s = y_{ij}^t \quad \forall s, t \in S, s \neq t \quad (3.25)$$

$$y_{ij}^s \in \{0,1\} \quad \forall (i,j) \in A, \forall s \in S \quad (3.26)$$

$$x_{ij}^k \geq 0 \quad \forall (i,j) \in A, \forall k \in K, \forall s \in S \quad (3.27)$$
where $p_s$ is the probability of realization of each scenario $s \in S$, and superscript $s$ for each decision variable indicates that variable, if any specific scenario $s \in S$ is observed. It should be noted that the flow capacity constraints link the first and second stage variables, and therefore, it is not possible to have a scenario separable problem. To overcome this difficulty a copy of the first stage variable can be considered for each scenario as presented in the above model through the third (i.e., nonanticipativity constraints) and fourth constraints. A single specific design is achieved if all scenario designs are equal. As it was pointed out, at this stage if the nonanticipativity constraints are relaxed, scenario decomposition across the problem is possible. However, the number of nonanticipativity constraints might be very large considering the number of scenarios. To rectify this issue, it was proposed to define an overall design vector $\tilde{y}_{ij} \in \{0,1\}, \forall (i,j) \in A$, and replace the following constraints for the nonanticipativity constraints in the original problem.

$$y_{ij}^s = \tilde{y}_{ij} \quad \forall (i,j) \in A, \forall s \in S, \quad (3.28)$$

$$\tilde{y}_{ij} \in \{0,1\} \quad \forall (i,j) \in A. \quad (3.29)$$

The above constraints indicate that each scenario design should be the same as the overall design and also imposes integrality conditions over the overall design. With this new formulation of the problem, now it is possible to penalize each scenario separately when the nonanticipativity constraints are relaxed (i.e., Lagrangian relaxation is applied). By relaxing the nonanticipativity constraints using an augmented Lagrangian method and considering the binary requirements of the design variables, the objective function of the main deterministic problem is:

$$\min \sum_{s \in S} p_s \left( \sum_{(i,j) \in A} \left( f_{ij} + \lambda_{ij}^s - \rho \tilde{y}_{ij} + \frac{\rho}{2} \right) y_{ij}^s + \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^{ks} \right)$$

$$- \sum_{(i,j) \in A} \lambda_{ij}^s \tilde{y}_{ij} + \sum_{(i,j) \in A} \frac{1}{2} \rho \tilde{y}_{ij} \quad (3.30)$$
where $\lambda_{ij}^s, \forall (i,j) \in A$ and $\forall s \in S$, are the Lagrangian multiplier elements and $\rho$ is the penalty terms for the relaxed constraints. However, the relaxed problem is not yet scenario separable. It can be decomposed into the sub-problems if the overall design is considered to be a fixed given value vector. Consequently the sub-problems can be represented as follows:

$$
\begin{align*}
\min & \quad \sum_{(i,j) \in A} \left(f_{ij} + \lambda_{ij}^s - \rho \bar{y}_{ij} + \frac{\rho}{2} \right) y_{ij}^s + \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^{ks} \\
\text{s.t.} & \quad \sum_{j \in N^+(i)} x_{ij}^{ks} - \sum_{j \in N^-(i)} x_{ji}^{ks} = d_i^{ks} \quad \forall i \in N, \forall k \in K, \forall s \in S \\
& \quad \sum_{k \in K} x_{ij}^{ks} \leq u_{ij} y_{ij}^s, \quad \forall (i,j) \in A, \forall s \in S \\
& \quad y_{ij}^s \in \{0,1\} \quad \forall (i,j) \in A, \forall s \in S \\
& \quad x_{ij}^{ks} \geq 0 \quad \forall (i,j) \in A, \forall k \in K, \forall s \in S.
\end{align*}
$$

(3.31)

(3.32)

(3.33)

(3.34)

(3.35)

The multiplier elements and the penalty terms are being used in an iterative fashion to panelize all design variables in different scenarios. This is done in order to make the difference between each scenario design variable and overall design as minimum as possible. As soon as the multiplier and penalty adjustments are determined, the above sub-problems become determinist problems, and by adjusting these two terms, it is possible for all scenario design variables to converge to the overall design $\bar{y}_{ij} \in \{0,1\}, \forall (i,j) \in A$. One of the main advantages of transforming the original problem into the deterministic problem is the luxury of using more sophisticated and efficient algorithms available for solving this type of problems. Nevertheless, how to have an overall design for the reference point, and how to achieve consensus among all different scenarios are a major step in designing any solution algorithm for solving sub-problems.

By solving the model for sub-problem, it is then possible to generate a general design using aggregation operator on $y_{ij}^s$ at each iteration. The aggregation operator is a function that can simply aggregate different references obtained through each sub-
problems or scenario to a single response. This be can achieved through considering appropriate weights for each scenario reference obtained from available information. The aggregation operator defined by Rockafellar et al. [38] was average function. For the CMND problem, the weights were considered to be the probability of each scenario, and therefore, the aggregation operator can be defined as:

\[
\bar{y}_{ij}^\nu = \sum_{s \in S} p_s y_{ij}^{\nu_s}, \quad \forall (i,j) \in A
\]  

(3.36)

where \( \nu \) is the index count for the iteration number. It was however pointed out that this aggregation operator might not produce a general feasible solution to the problem. This is resulted from the fact that if there is no single consensus among all designs obtained from each scenario, \( \bar{y}_{ij}^\nu \) does not belong to \( \{0, 1\} \). Therefore in this situation, the integrality constraint is violated. Using the proposed aggregation function for the non-convex problem, there is no guarantee for the convergence of the overall solution. For the CMND problem, although this operator may not provide any good or feasible solution, it was suggested that it can be a good guidance for the overall solution strategy, and also indicates the general trend among all designs obtained from sub-problems. Considering this fact, it was proposed that the abovementioned aggregation operator can be used for the first stage search. The obtained results can then be used as a guideline for the second stage search to achieve a single consensus among all designs of sub-problems. To generate a feasible solution, it was also proposed to use a worst-case analysis in which all open arcs obtained from each scenario solution are considered in the overall solution. Although, this guarantees to generate a feasible solution, it mainly overestimates the open arcs that should be included in the overall design when consensus is low among the designs of different sub-problems.

Another important aspect is penalty adjustment. According to the proposed solution strategy proposed in the original work performed by Crainic et al. [43], to achieve consensus among the sub-problems, the fixed cost should be adjusted through iterative process using multiplier elements, and penalty terms. This is performed in order to minimize the difference between scenario designs and overall design. After applying PH
strategy to the CMND problems, the Lagrangian multiplier elements and penalty terms can be updated as well.

This guarantee that the penalty terms approach to infinity as the number of iteration increases. It was mentioned that there are three possible situations for adjustment following this method. If the design variables of the current and past iterations are equal, the fixed cost remains unchanged. If the current design variable is smaller than its counterpart from previous iteration (i.e., \( y_{ij}^{sp} < \tilde{y}_{ij}^{p-1} \)), the fixed cost related to this design variable is decreased in the objective function. Consequently, the fixed cost of a closed arc in a sub-problem due to non-consensus is adjusted in a way that the arc becomes open in the related scenario. The final case is that the current design variable is larger than its counterpart from previous iteration. For this situation, the associated fixed cost in the objective function of the sub-problem is increased and in an opposite fashion to the previous case, the open arc will be closed.

The aforementioned method developed based on the PH algorithm is used as the basis for proposing another efficient new heuristic penalty adjustment. This method in each iteration targets those arcs with the high level of consensus. As introduced before, the overall design can represent the trend of the sub-problem designs. A high value of this term for a specific arc indicates that a large number of that arc are open among scenario sub-problems whereas the low value of this term shows that a small number of this particular arc are open among different scenarios. Consequently, this can be a motivation that if the value of the overall design of a specific arc is less than a threshold, the fixed cost associated with that arc can subsequently be increased in order to force sub-problems to avoid using that arc. Conversely, if the value of the overall design is more than a specific threshold, then by decreasing the associated fixed cost, it is possible to force the sub-problems to use that specific arc. In this method as the fixed cost is adjusted for all sub-problems, it can be considered a global adjustment. With this approach, it is also possible only to target some of the sub-problems which required modification. For this purpose, the different between the values of the design decision of a sub-problem and the overall design decision is studied. If this value is large, it is then
possible only to adjust the fixed cost of that specific sub-problem in the upcoming iteration. In this regards as well as computational experiments, more details are provided in the related original work [43].

In PH algorithm, the performance of the algorithm is highly sensitive to the choice of the penalty parameter. There is not any specific approach for the penalty adjustment strategy; the proper strategy is mainly defined based on the empirical analysis as well as the nature of the problem. Concepts supported by theory indicate that the higher values of penalty parameter should produce a faster convergence. However, higher value of the penalty parameter does not necessarily reduce the number of iterations. Large penalty parameter values result in a faster convergence in the dual variables and a slower convergence in the primal variables. Therefore, some intermediate range of values can be more efficient for improving the overall algorithm performance. In addition, although finding an effective range for the penalty parameter entails empirical experiments, it is not sensitive to the number of the sub-problems [38, 39, 41]. For integer cases, there are not any reports on the theoretical principles for the convergence of the algorithm. Heuristic strategy normally terminated by the following causes, CPU limitation, a certain number of iteration without any significant improvement or any other conditions defined based on the structure of a problem. However, in this type of circumstances, the result might not be the solution over which all sub-problems are on consensus [43].

3.4 Stochastic network flow mixed-integer programming for financial planning

In this section, development of a multistage stochastic mixed-integer programming approach for financial planning using network flow structure is described. This approach is intended to capture different aspects of a comprehensive portfolio management strategy including holding a small number of securities, and consequently minimizing the total number of transactions in order to reduce the transaction costs. This can be achieved by considering cardinality constraint by which it is possible to control the number of securities in the portfolio, or by imposing restrictions on the transaction costs directly through limiting the securities purchased and sold. It should be noted that
considering this constraint increases the level of complexity associated with the solution process [12,14,24]. In addition, periodically revision of a designed portfolio throughout its investment time horizon is of critical importance due to the presence of some uncertainties such as expectations about the future returns or transaction costs as the market environment changes day by day. It should also be considered that during the investment time horizon, new cash flows might become available for the portfolio investment, either investing in the same available securities or new securities. Trading size also is another feature of a comprehensive portfolio design process, both at the time of building a new portfolio or rebalancing an existing one. This feature can be captured within the model formulation by imposing lower and upper bound restrictions on the securities included in the portfolio.

In a network flow structure, as described in the preceding sections and shown in Figure 3.4, each node is associated with an asset at each stage and also each arc of the network indicates a transaction decision which can be made at each time stage as well as scenario. It was also pointed out that each arc or transaction decision normally has a multiplier that can scale the assets flowing through each arc. This multiplier here is represented by transaction costs determined at each stage and scenario, considering the rules and regulations of the market at that particular condition. In addition, in this problem the arcs between two stages are scaled by rate of returns.

In this proposed approach, without loss of generality if a basic two-stage model is considered, the first stage is designed for deposition of initial capital in the selected securities included in the investment portfolio. The second stage can then be applied for rebalancing the portfolio considering the transaction costs as well as rate of returns for selected securities. As it is evident, the first stage decisions are made before the random events are taken place and also influence the parameters defined for the model. The second stage decisions are then made under the light of the realization of the random event to rebalance the portfolio designed on the bases of the first stage decision. In another word, the first stage decisions are fixed, the random events take place, parameters for building the portfolio become known, and then the recourse actions are defined through the second stage decisions. The objective function of the problem to be
optimized is intended to be the total Portfolio value. The total portfolio value is designed to be maximized so that the transaction costs should obviously be kept as minimum as possible throughout maximizing the portfolio value. To formulate a stochastic network flow mixed-integer programming for financial planning problems, the same process described in this chapter so far should be followed. For this purpose, the network \( G = (N, A) \) is defined with the sets \( N \), and \( A \) for its nodes and directed arcs, respectively. \( S \) also defines a set of possible discrete scenarios for the random events. Using this set and under the realization of each scenario, one can approximate a stochastic problem by its deterministic equivalent problem. \( \tau \) indicates the decision periods in the planning horizon, \( \tau = \{ t: t = 0, \ldots, T \} \). Let \( P(s) > 0 \) be the probability of occurrence of each scenario \( s \in S \), and therefore, \( \sum_{s \in S} P(s) = 1 \). Obviously many scenarios may result in a set of identical values for the random events which is due to the same information history they have up to that time step. The progressively realization of the random event will be considered in the problem formulation. \( b_i^t \) for \( (i, j) \in A \) and \( t \in \tau \) is the external capital invested in security \( i \) at time step or stage \( t \), therefore, \( b_i^0 \) is the initial investment in security \( i \) at the beginning of the investment horizon. \( x_{ij}^{tt}(s) \) for \( (i, j) \in A \) and \( t \in \tau \) is represented the capital flowing through the arc \( (i, j) \) for the scenario \( s \in S \), and time step \( t \). Considering the notation, \( x_{ij}^{11}(1) \) means that the cash flow across arc \( (i, j) \) for the first scenario and at stage 1, and \( x_{ii}^{01}(1) \) means that the cash invested in security \( i \in N \) carried forward from stage zero to the next consecutive stage for the first scenario. \( T_{ij}^{tt} \) for \( (i, j) \in A \) and \( t \in \tau \) denote the transaction cost with the same notation used for super and subscript of \( x_{ij}^{tt}(s) \). Let \( R_i^{(t-1)t} \) for \( i \in N \) and \( t \in \tau \) be the return of the security \( i \) for the time period between stage \( t - 1 \) and \( t \). Finally, \( A^{tt} \) for \( t \in \tau \) denotes the set of arcs in the time step \( t \). Therefore, the objective function as mentioned is to maximize the total portfolio value which can be achieved by minimizing the following terms:

\[
\min_{x, p} \sum_{(i, j) \in A, i \neq j} T_{ij}^{00} x_{ij}^{00} - \sum_{t=1}^{T} \sum_{s \in S} P^t(s) \left( \sum_{(i, j) \in A^{tt}(s)} (1 - T_{ij}^{tt}(s)) x_{ij}^{tt}(s) \right)
\]  

(3.37)
The first summation term is minimizing the transaction cost while the second summation term is defined to minimize the second stage transaction cost while maximizing the total value of the portfolio.

To construct a portfolio which satisfies the main desired investment goals, it is required to impose proper constraints on the objective function. Solution of a model in a general formulation may result in a portfolio with a large number of securities with small positions. This type of portfolios is not desirable as a large number of securities in a portfolio implies a large transaction cost each time that the portfolio is rebalanced. In addition, managing this type of portfolios is relatively costly due to their size and required monitoring process. To avoid this drawback while having a portfolio with a proper diversity, the total number of asset in the portfolio should be restricted with a specified number. In order to control the number of securities to hold in the portfolio, a binary variable $y_t$ is introduced. Therefore to select a security $i \in N = \{i: i = 1, ..., n\}$, $y_t$ should be defined as:

$$y_t = \begin{cases} 1, & \text{if security } i \text{ is included in the portfolio} \\ 0, & \text{otherwise} \end{cases} \quad (3.38)$$

where $y_t \in \{0,1\}$ as pointed out is a binary variable. Using this variable the number of securities in a portfolio at each time step can be controlled by the following constrains.

$$\sum_{i \in N} y_{it}^t = K \quad \forall \ t \in \tau. \quad (3.39)$$

The next step is to consider the network flow conservation constraints that indicate the input and output flows on all nodes at each time steps should be in balance and do not change. Therefore, considering outward and inward flows into each node the conservation constraints for the first and subsequent higher stages can be written as:

$$\sum_{(i,j) \in A_0^0} x_{ij}^0 - \sum_{(j,i) \in A_0^0, j \neq i} x_{ji}^0 \leq b_i^0 \quad \forall \ i \in N, \quad (3.40)$$
The first flow conservation constraints are defined for the first stage nodes and the second one for upper stages. Since for the first stage flow the transaction costs are not directly considered in these constraints the equality signs are replaced by inequality to not violate the constraints. Finally, the capacity constraints on the network by which the upper and lower bounds on the flows or variables are defined should be designed. If the arc \((i,j) \in A\) is selected in the network then its total amount of flow through this arc cannot exceed the predefined lower and upper bound capacities, \(l_{ij}^{tt}\) and \(u_{ij}^{tt}\) for \((i,j) \in A\) and \(t \in \tau\), respectively. The followings define the lower and upper bound constraints for the first and other subsequent stages, respectively.

\[
\sum_{(i,j) \in A^{tt}(s)} x_{ij}^{tt}(s) - \sum_{(j,i) \in A^{tt}(s)} (1 - T_{ji}^{tt}(s))x_{ji}^{tt}(s) \leq R_i^{(t-1)t}(s)x_{ii}^{(t-1)t}(s) \forall i \in N, s \in S, t \in \tau.
\]

The first flow conservation constraints are defined for the first stage nodes and the second one for upper stages. Since for the first stage flow the transaction costs are not directly considered in these constraints the equality signs are replaced by inequality to not violate the constraints. Finally, the capacity constraints on the network by which the upper and lower bounds on the flows or variables are defined should be designed. If the arc \((i,j) \in A\) is selected in the network then its total amount of flow through this arc cannot exceed the predefined lower and upper bound capacities, \(l_{ij}^{tt}\) and \(u_{ij}^{tt}\) for \((i,j) \in A\) and \(t \in \tau\), respectively. The followings define the lower and upper bound constraints for the first and other subsequent stages, respectively.

\[
l_{ii}^{00}(y_i^{00}) \leq x_{ii}^{00} \leq u_{ii}^{00}(y_i^{00}) \forall i \in N, \quad (3.42)
\]

\[
l_{ij}^{tt}(s)(y_i^{tt}) \leq x_{ij}^{tt}(s) \leq u_{ij}^{tt}(s)(y_i^{tt}) \forall i \in N, (i,j) \in A^{tt}(s), s \in S. \quad (3.43)
\]

In these constraints, variable \(y\) is considered in order to impose the lower and upper bounds on the arcs which are included in the design. For the first stage in which the securities in the portfolio are selected, the lower and upper bounds are imposed on the nodes without compromising the design of the network.

To solve this model in order to analyze its introductory performance, the model is design and solve for two-stage stochastic mixed integer network, however as pointed out this approach can be extended for the multistage portfolio management. Figure 3.5 depicts the structure of a two-stage network flow with four nodes at each stage. In this figure the multiplier for each arc (i.e., the transaction costs and returns) are shown in the parenthesis. Considering all description for the problem presented in this section as well as the structure of the network schematically shown in Figure 3.5, the model formulation for the two-stage problem can be written as follows:
\[
\min_{x,p} \sum_{(i,j) \in A^0, i \neq j} T_{ij}^{00} x_{ij}^{00} - \sum_{s \in S} P(s) \left( \sum_{(i,j) \in A^1(s)} (1 - T_{ij}^{11}(s)) x_{ij}^{11}(s) \right) \\
\text{s.t.} \quad \sum_{(i,j) \in A^0} x_{ij}^{00} - \sum_{(j,i) \in A^0, j \neq i} x_{ji}^{00} \leq b_i^0 \quad \forall i \in N,
\]

\[
\sum_{(i,j) \in A^1(s)} x_{ij}^{11}(s) - \sum_{(j,i) \in A^1(s)} (1 - T_{ij}^{11}(s)) x_{ji}^{11}(s) \leq R_{ij}^{01}(s) x_{il}^{01}(s) \quad \forall i \in N, s \in S,
\]

\[
\sum_{i \in N} y_{i}^{00} = K, \quad y_{i} \in \{0,1\}
\]

\[
l_{ii}^{00}(y_{i}^{00}) \leq x_{ii}^{00} \leq u_{ii}^{00}(y_{i}^{00}) \quad \forall i \in N,
\]

\[
l_{ij}^{11}(y_{i}^{11}) \leq x_{ij}^{11}(s) \leq u_{ij}^{11}(y_{i}^{11}) \quad \forall i \in N, (i,j) \in A^1(s), s \in S.
\]

**Figure 3.5** Two-stage network with four nodes at each stage
In the model, the initial investment for each security $b_i$, lower and upper bounds for the first and second stages $l_{ii}, l_{ij}, u_{ii}$ and $u_{ij}$ are constant parameters, positive, and larger than zero. Uncertainty is defined through the arc multipliers, the return of each securities $R_i$ as well as the transaction costs $T_{ij}$. The amount of investment in each securities, $x_{ij}$ for the first and second stages, and also the binary variable defined for the cardinality constraint are the design variables for the designed two-stage model.

### 3.5 Numerical results and analysis

In this section, different aspects of the proposed approach described in the last section are presented through different numerical examples. For this purpose and to perform the numerical analysis, the input parameters including the transaction costs and rate of returns are obtained through random data generation. This is done in order to reflect the uncertainties associated with different generated scenarios throughout the investment time horizon. The developed model is coded in MATLAB (www.mathworks.com) and solved using Gurobi optimizer 5.6 (www.gurobi.com). All computations are performed using a computer with Intel Core(TM) i7-3520M CPU@ 2.90GHz processor and 8 GB RAM.

The following numerical example shows the modelling results for the design of a portfolio consisting of two securities, $K = 2$, selected from a financial market with total four securities, $n = 4$. For this financial planning three scenarios, $s = 3$, are generated. The values generated for the returns, transaction costs for the first stage, and transaction costs for the three generated scenarios of the second stage are shown in Table 3.1, Table 3.2 and Table 3.3, respectively.

<table>
<thead>
<tr>
<th>Table 3.1 Return values for three scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_i(s)$</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>$s=1$</td>
</tr>
<tr>
<td>$s=2$</td>
</tr>
<tr>
<td>$s=3$</td>
</tr>
</tbody>
</table>
Table 3.2 Transaction cost for the first stage

<table>
<thead>
<tr>
<th>$T_{ij}^{00}$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.05</td>
<td>0</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.05</td>
<td>0.05</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.3 Transaction cost for the second stage with three scenarios

**First scenario**

<table>
<thead>
<tr>
<th>$T_{ij}^{11}(1)$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0793</td>
<td>0.0788</td>
<td>0.0780</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0784</td>
<td>0</td>
<td>0.0757</td>
<td>0.0790</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0837</td>
<td>0.0787</td>
<td>0</td>
<td>0.0760</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0825</td>
<td>0.0794</td>
<td>0.0810</td>
<td>0</td>
</tr>
</tbody>
</table>

**Second scenario**

<table>
<thead>
<tr>
<th>$T_{ij}^{11}(2)$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0518</td>
<td>0.0539</td>
<td>0.0484</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0451</td>
<td>0</td>
<td>0.0522</td>
<td>0.0502</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0538</td>
<td>0.0547</td>
<td>0</td>
<td>0.0529</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0471</td>
<td>0.0500</td>
<td>0.0510</td>
<td>0</td>
</tr>
</tbody>
</table>

**Third scenario**

<table>
<thead>
<tr>
<th>$T_{ij}^{11}(3)$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0170</td>
<td>0.0238</td>
<td>0.0169</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0184</td>
<td>0</td>
<td>0.0155</td>
<td>0.0152</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0237</td>
<td>0.0185</td>
<td>0</td>
<td>0.0196</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0159</td>
<td>0.0222</td>
<td>0.0151</td>
<td>0</td>
</tr>
</tbody>
</table>
The transaction costs are considered to be the total amount of the costs for buying or selling different securities as well as the other related portfolio management costs. The values of these costs are generated according to the market situation which is shown through the interest rates. It means that the transaction costs are dependent to the scenarios. In addition, the initial investment for the portfolio is 400 dollars, therefore $b_l^0$ is set to 100 dollars for each securities at the first stage. The final results for this example for the first stage variables, $x_{i,j}^{00}$, and second stage variable, $x_{i,j}^{11}(s)$ for $s = 1,2,3$, are respectively:

$$
\begin{align*}
x_{i,j}^{00} &= \begin{bmatrix} 0 & 0 & 100 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 300 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}, &
x_{i,j}^{11}(1) &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 462.375 & 100 \\ 0 & 0 & 100 & 103.96 \end{bmatrix} \\
x_{i,j}^{11}(2) &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 299.318 & 100 \\ 0 & 0 & 100 & 97.082 \end{bmatrix}; &
x_{i,j}^{11}(3) &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 151.465 & 100 \\ 0 & 0 & 100 & 87.5192 \end{bmatrix}
\end{align*}
$$

These results for the first stage and second stage including three generated scenarios are schematically shown in Figure 3.6. In the figure, the transaction costs for the second stage and returns of the securities transferred from first to the second stage are also shown in the parenthesis. As seen in the figure and shown in the aforementioned first stage result, $x_{i,j}^{00}$, for a portfolio including only two securities the capital invested in the first and second securities are transferred into the third security. Therefore, in the first stage the portfolio is designed with two securities corresponding to the node three and node four of the network. In the second stage, there are three scenarios, as depicted in Figure 3.6, in which the portfolio is rebalanced according to the values of the random variables that influence the value of a set of parameters defined within the model. These parameters were defined as listed in Table 3.1, Table 3.2, and Table 3.3.
To show more explicitly the role of the cardinality constraint in a realistic portfolio selection and management, the same problem presented in this section is solved again but with different number of securities included in the designed portfolio, $k = 3$. For this case study, Table 3.4 and Table 3.5 provide the return values of the three generated scenarios and their corresponding transaction costs, respectively. The initial investment is considered the same as set for the first case study and shown in Figure 3.6.

**Table 3.4** Return values generated for three scenarios

<table>
<thead>
<tr>
<th>$R_i(s)$</th>
<th>$n = 1$</th>
<th>$n = 2$</th>
<th>$n = 3$</th>
<th>$n = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S=1$</td>
<td>1.9330</td>
<td>1.9031</td>
<td>1.5685</td>
<td>1.1327</td>
</tr>
<tr>
<td>$S=2$</td>
<td>0.9815</td>
<td>0.9777</td>
<td>1.0052</td>
<td>1.0247</td>
</tr>
<tr>
<td>$S=3$</td>
<td>0.0598</td>
<td>0.0243</td>
<td>0.5110</td>
<td>0.8834</td>
</tr>
</tbody>
</table>
Table 3.5 Transaction cost for the second stage with three scenarios

<table>
<thead>
<tr>
<th></th>
<th>$T_{ij}^{(1)}$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0789</td>
<td>0.0829</td>
<td>0.0755</td>
<td></td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0838</td>
<td>0</td>
<td>0.0804</td>
<td>0.0780</td>
<td></td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0759</td>
<td>0.0784</td>
<td>0</td>
<td>0.0754</td>
<td></td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0843</td>
<td>0.0823</td>
<td>0.0839</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$T_{ij}^{(2)}$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0457</td>
<td>0.0504</td>
<td>0.0493</td>
<td></td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0522</td>
<td>0</td>
<td>0.0548</td>
<td>0.0497</td>
<td></td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0537</td>
<td>0.0530</td>
<td>0</td>
<td>0.0506</td>
<td></td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0508</td>
<td>0.0478</td>
<td>0.0533</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$T_{ij}^{(3)}$</th>
<th>$j = 1$</th>
<th>$j = 2$</th>
<th>$j = 3$</th>
<th>$j = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0163</td>
<td>0.0228</td>
<td>0.0205</td>
<td></td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0200</td>
<td>0</td>
<td>0.0216</td>
<td>0.0180</td>
<td></td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0214</td>
<td>0.0186</td>
<td>0</td>
<td>0.0243</td>
<td></td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0180</td>
<td>0.0228</td>
<td>0.0152</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The final results for the first stage variables, $x_{ij}^{00}$, and the second stage variable, $x_{ij}^{11}(s)$ for all three scenarios are computed as follows. In addition, Figure 3.7 schematically shows the final results for the three generated scenarios of the designed portfolio consisting of three securities.
As seen in the figure, for this case study the designed portfolio with three securities consists of securities that are represented by the first, third and fourth nodes in the network structure. Comparing these two cases more precisely, the first case was solved
again with the same return values considered for the second case. The total final investments are almost the same, but the final total value of the portfolio with three assets is slightly smaller than that of the portfolio with two assets (i.e., 5.94 per cent). However, the second portfolio is more diversified. A well-diversified portfolio has relatively a reduced risk exposure throughout the investment horizon. In comprehensive and realistic financial models appropriate constraints are considered to impose diversification or tradability of the portfolio [2]. As described in the second chapter, one approach to consider the diversification in a portfolio and consequently minimize the associated risk is by using variance. Therefore, the developed modelling algorithm presented in this chapter has the capability of designing a well-diversified portfolio according to the defined goals of the investors. The model has $n^2$ continuous and $n$ binary variables at the first stage, and $n^2 \times s$ continuous variables at the second stage, where $n$ and $s$ are the number of nodes and scenarios, respectively. Therefore, the total continuous and binary variables is $n^2(1 \times s) + n$. Figure 3.8a and b show the number of variables versus the number of securities for different number of scenarios.

The proposed model also has total $3 \times n + n \times s(1 + 2s) + 1$ constraints. Figure 3.9 shows the number of constrains versus the number of scenarios and securities for $s = 1, ..., 19$ and $n = 1, ..., 500$. Therefore, for the mode of the second case study, with four nodes, three scenarios and a portfolio with three securities, there are 68 variables and 121 constraints. To show the performance of the model solving with the commercial solver, Figure 3.10 shows the gap of the final solution for a model with ten nodes for different cardinality constraints. This figure indicates the role of the cardinality constraints on the solution process, and also emphasizes the fact that to have an efficient solution process, the problem specific algorithm should be designed. In this regard, the solution time also should be taken into consideration a well. For instance, for the model with 10 nodes, 3 scenarios and cardinality constraint equal to 8, the solution time is 0.02 seconds while this number for the problem with same characteristics but 20 nodes is 1002.76 seconds.
Figure 3.8 Number of variables based on the numbers of securities and scenarios: a) $s=1,\ldots,10$ and b) $s=10,20,\ldots,100$. 
Figure 3.9 Number of Constraints based on the numbers of securities and scenarios

Figure 3.10 Performance of the model using the commercial solver
Chapter 4

Index tracking

The passage of time for any investment is associated with uncertainty regarding the future. As discussed in the preceding chapters this fact plays a determining role in any portfolio optimization strategies. Any proper optimization method should be capable of making proper decision in the face of unexpected events or uncertainties. In financial markets, market indices are measures of the market trends and its circumstances. Index is an indicator that investors look at to understand how the market is doing at large. As pointed out, market indices are an important tool in financial decision making for the investor as these indices show the performances and trends of a large section of the market. For instance, the S&P 500 (i.e., the standard & Poor’s 500) is defined based on the market capitalizations of 500 large companies. Therefore, using the market indices, portfolio managers can evaluate the performance of their portfolio based on the performance of the market. This portfolio management strategy is an excellent candidate for investments such as pension funds, corporate financial obligations, and insurance premiums. For instance, corporate pension plans in the USA have invested over a quarter of their equity holdings in index funds considering this fact that stock market indices have historically shown growth in the long term time period [44,45,46].

4.1 Active and passive portfolio management

Active and passive portfolio managements are two different approaches to manage portfolios. In active portfolio management, different investment strategies are employed
to manage a portfolio with the intention of receiving higher returns than the market. As pointed out in the literature, many investors manage their investments against a benchmark using an active strategy. This is conducted by a given percentage to outperform the benchmark subjected to a constraint on tracking error variance. By practicing this method, one has to create a portfolio that on one hand is mean-variance inefficient, and on the other hand has associated risk larger than 1 compared against the benchmark within a set of certain conditions. However, it is obvious that in some cases large losses have been resulted from these portfolios [47, 48]. Consequently, the main problem in active portfolio management strategy is to determine the circumstances of the added value to the portfolio against the risk undertaken. This problem is more prominent when the portfolios have the performance fees. The performance fee can simply be defined as a fee that is charged by a portfolio manager for generating positive returns. Active portfolio management with performance fees can be considered as an option by which a manager is interested to take more risks for generating larger increase in the value of the option. To manage this issue, as pointed out a tracking error variance or tracking error volatility also known as TEV is imposed by which a limit is defined on the deviation of the portfolio from the corresponding benchmark [49]. Two strategies defined for overcoming the aforementioned associated problem with active portfolios, one method constrained the portfolio beta proposed by Roll [50], while in another method developed by Jorion [49] a restrict was imposed on the variance of an investment portfolio. Alexander et al. [48] developed another strategy in which the Value-at-Risk is constrained to properly manage active portfolios.

While, an active portfolio is designed to outperform the market, a passive portfolio management constructs a portfolio that closely mimics the market trends instead of outperforming them. The nature of these two methods imply the fact that contrary to the active portfolio managers who have obviously more freedom to apply their experiences and expertise judgments to build portfolios, passive portfolio managers hold much less flexibility. The passive portfolio managers should generate a portfolio within a defined framework and strictly follow a certain set of criteria, on the other hand an active portfolio manager as pointed out can use his or her intuition developed through
years of experience and training to pick winner securities whose added values to the portfolio can beat the benchmark. Based on the passive management definition, a fund managed to produce the return similar to that of the index is called an index fund or a tracker fund. The market indices are available for both the global and regional market performances. For instance, FTSE All World Index and S&P 500 (SPX) are two examples for the global and regional indices, respectively. In order to generate and subsequently maintain a portfolio within the framework of passive or active portfolio management strategies, first the main advantages of these two methods along with their disadvantages should carefully be understood. Afterwards, it is even possible to define a hybrid method to encompass the advantageous features of both passive and active strategies alongside each other, and also have a minimum impact of their drawbacks.

Portfolios by active management strategy cannot simply outperform the market in practice. In addition, active management strategy has high fixed costs resulted from cost of the management team as well as regular trading costs, which is a considerable cost resulting from the frequent security picking. Due to the exclusive specifications of the passive strategy, it is an excellent candidate for investments such as pension funds, corporate financial obligations, and insurance premiums. For instance, corporate pension plans in the USA have invested over a quarter of their equity holding in index funds considering this fact that stock market indices have historically shown growth in the long term time period \[44, 45, 46\]. It should be noted that the passive strategy therefore is a method that many portfolio managers use to create models for constructing index funds and it is mostly intended for the long term wealth acquisition. However, the main disadvantage of passive strategy is that once the market indices fall, consequently the return of the portfolios generated within the framework of the passive management fall correspondingly. Another point is the risk associated with a portfolio, in passive management the investment is only affected by the risk related to the market. For the investment managed with active strategy, the risk factor associated with the company also exists in addition to the risk related to the market. Historical data also reveals that the active management does not outperform the market for relatively long term investments. In addition, considering the fact that the indices have increased over years,
managing a portfolio using passive strategy can provide reasonable return without engaging the portfolio with unnecessary risk [2, 44, 46,51,52].

A passive managed portfolio can simply be generated by picking and consequently purchasing all securities included in the index which is intended to be tracked. In this way, the investment portfolio also known as complete replication portfolio can exactly reproduce the index trend throughout the investment time horizon. However, this simple approach associated with several conceptually as well as computationally complications. Beasley et al. [46] addressed some these problems. It was pointed out that some of the securities in the index might exist in a very small numbers, therefore, it is difficult and usually expensive to include and maintain this type of securities into the portfolio during the investment period. This situation most probably occurs when the index is made of a large number of securities. Another important problem associated with this type of portfolio arises from the fact that over a long time period there is a high probability for the index going through several changes due to the changes in the composition of its underlying securities. Consequently, each time that the composition of the index is changed, inevitably the composition of the investment portfolio must also be modified accordingly, based on the new added securities and their corresponding weightings. There are several reasons for the underlying composition of an index to be changed. One is the merger. Several mergers of different companies might happen during any investment. Another reason is that one or more companies might lose their merit to be included in the index due to the drop of their prices. Every time in this type of situation, new eligible companies will be replaced by those lost their merits in the index, and therefore, the related securities in the portfolio must be altered accordingly. The final problem associated with passive management of the complete replication portfolios is related to the transaction costs. To have an optimum and robust portfolio management the transaction costs in the course of investment for both selling and purchasing different securities should be carefully monitored and constrained. For this type of portfolios, there is no limit for the transaction costs, and therefore it might have a significant adverse effect on the added value of the portfolio. Considering all the aforementioned problems and their circumstances accompanied by the complete
replication, most passively managed portfolios comprise relatively fewer securities than of those included in the index. This might make the index tracking problems more complex however it does have a relatively crucial positive impact on the portfolio’s added value. The index tracking problem therefore can be define as the design of a portfolio with a particular number of securities which can replicate the index performance. For more detail in this regard one can consult with the literature [44-54].

4.2 Index tracking models

An index in order to be a proper tool to support investors for making decision should have a set of characteristics regardless of their types. An index should be chosen in a way that estimates the universe of securities a manager invests in. In addition, to realistically survey the market performance, it is appropriate that an index forms based on the traded issues associated with the investments. Investor should be able to replicate the market performance using the index (i.e., market information), and for this purpose, a relatively stable composition of the index is also required. To create an index, the components of the index should first be selected, and then the important step is to define the role of each component or constituent in the formation of the index. An index might be formed using equally weighted components or securities. Sometimes it is better to have price weighted or cap weighted approaches to define an index as well. As the name of the first type, equally weighted implies, all securities are given the same weight to from an index in this format. On another hand, in the price weighted approach, only prices of one share of securities, forming an index, are used. In the other approach, the weights are allocated to the securities considering their price as well as their available number of shares. As it can be noted, in the price weighted approach, the index is more affected by more expensive securities. The return of an index is therefore defined based on the returns of its securities, which is normally their weighted returns. More details in this regard can be found in the literature [2].

As mentioned, a portfolio which its performance or its growth rate is intended to replicate the performance of an index is called an index fund. Index fund can be modelled based on two different approaches, structural and co-movements. In the former
In the co-movement approach, on the other hand, the portfolio is generated in a way that its response to the various potential risk factors should be similar to the response of the index regardless of the risk contents of the designed portfolio. In the former approach, the universe of available securities is grouped into different cells or sectors based on a common characteristic. According to this method, the portfolio is then generated by investing in the securities from different cells, and therefore, the difference between index cell holding and portfolio is used to measure how approximately a portfolio is tracking an index. One can consult with the literature for more detail [2,44,53].

The optimization models for index funds are developed and formulated following the developments of MVO and CAPM strategies [46]. Using CAPM strategies, Rudd [53] listed the requirements for an index fund as: beta should be equal to one relative to the index, factors which are common for the index and portfolio should be as close as possible, and finally the index fund should have the minimum specific risk. The first two conditions define the objective function which is minimizing residual risk, and as pointed out the rebalancing is performed whenever related constraint for transaction cost is violated or beta is deviating from unity. He developed a factor model in which the objective function was defined to consider the transition costs. This is done through a weighting parameter in order to remove this cost considering an initial tracking portfolio. The model was built assuming that the transaction costs have no limit and are separated from the capital intended for the investment in the tracking portfolio. It was pointed out that incorporating the transaction cost into the objective function of the optimization model provides the opportunity of quantifying the effect of adding or removing a security from the portfolio directly. Consequently, it is possible to compare the benefit of investing in a particular security with its related transaction costs. This work was further developed by other research groups [55]. In this regards many research groups also used MVO modelling approach to develop an index tracking model. For instance, one of the first works in this aspect was performed by Hodges [56] in which he analyzed and compared the trade-off curves of variance and return for a standard MVO
model to those of the market index. Several other research programs were conducted using MVO model for index tracking problems [50, 57].

Zenios [2] presents indexation models designed based on the structural and co-movements strategies. A structural model for index tracking can be designed assuming that there are $K$ securities to invest in from an asset pool. It is required to determine the holding $x_i$ of securities $i = 1, ..., n$ in the portfolio selected from the asset pool. The weights of securities in the index that belong to each sector are determined from the index data considering $\sum_{q \in \text{segment}} \sum_{i=1}^{n} w_q$. Since the holdings $x_i$ are in percentage of total assets and as a security might belong to several sectors, the sectors should be normalized to add up to unity. The following optimization model generates a portfolio with a structure like the index structure:

$$\max F(x) \quad (4.1)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} P_{ij} x_i = w_j, \quad \forall j = 1, ..., K \quad (4.2)$$

$$\sum_{i=1}^{n} x_i = 1 \quad (4.3)$$

$$x \geq 0 \quad \text{and} \quad x \in G. \quad (4.4)$$

where $P$ is an indicator and $P_{ij} = 1$ if $i$ belongs to the $j$ sector otherwise 0, and also $G$ is the set of feasible solutions and might be constrained by more restrictions such as diversification constraints. In addition, for simplicity it is considered that the weight on the $l^{th}$ sector is $w_l$ by assuming that the number of existing securities in the index is the same as that of the sectors. For a tracking model using co-movement strategy, it should be noted that in the model using this approach, returns of the index as well as securities in the asset pool are considered uncertain. The tracking error for the discreet scenarios and assuming linear returns for both the index while the portfolio can be written as:
\[
R_{TE} = \sum_{i=1}^{n} r_i^s x_i - \sum_{j=1}^{K} w_j r_j^s
\]  
(4.5)

where \( s \in \text{scenario set } \Omega \), and \( r_i^s \) and \( r_j^s \) are the random returns of the portfolio and the index in scenario \( s \), respectively. The tracking model for this strategy is:

\[
\max \sum_{i=1}^{n} \bar{r}_i x_i
\]  
(4.6)

s.t. \( \sum_{i=1}^{n} r_i^s x_i - R_i \geq -e \), \( \text{for } s \in \Omega \)  
(4.7)

\[
\sum_{i=1}^{n} x_i = 1
\]  
(4.8)

\( x \geq 0 \) \( \text{and } x \in G \).  
(4.9)

Therefore, as described so far implicitly, the index tracking problem as a practical problem for passive portfolio management is to minimize the index tracking error considering a portfolio consisting of a particular number of assets. The advantage of a portfolio with a relatively small number of assets is the limited management and transaction costs. In a larger scope, the objective of tracking portfolios can simply be the index tracking error (i.e., as close as possible track an index), be a target return for the guaranteed investment product, or be the performance of relative and comparable competitors \([2,14,45]\). It should be also considered that portfolio management and consequently index tracking models can be either static or dynamic. In a dynamic approach, financial decision making problem can be conducted during the time horizon of the investment according to any newly arrived information. This is done through portfolio rebalancing. As also pointed out in the preceding chapters for the scenario based models, a dynamic strategy is taking into account a sequence of buy-sell, short term borrowing as well as lending decisions. Therefore, for a comprehensive tracking strategy characteristics of dynamic models should also be viewed. Along with this
purpose, important elements of a decision support system in which tracking portfolio is considered as a part of the whole system with specifically defined choices and compromises can be listed as follows [45, 58]:

- Benchmark: The benchmark or an index that a portfolio must track should be selected; portfolio and the selected index should have the same exposure to the economic sectors.
- System of risk measurement: It should be specified how the risk associated with the portfolio relative to the index is measured and tracked.
- Portfolio maintenance system: The price changes of a portfolio on the market should be considered through rebalancing strategy. In addition, a set of decision rules should be defined in order to consider any changes either to the market or to the portfolio composition caused by new issues, mergers or any other similar causes.
- Security number: Portfolio’s defined management strategy should be capable to balance the number of securities and the maximum allowable tracking error.
- Proper consideration for the generated cash flow: Cash flows generated by the portfolio during the investment time horizon should be designed through a set of decision rules.

A brief review on studies in which some of the abovementioned characteristics of the index tracking models were considered is available in the literature [14,45,58]. However, it should be considered that the index tracking portfolio problem with cardinality constraints (i.e., constraint on total number of assets) is NP-hard problem and therefore, to solve this type of problems using heuristic methods are highly recommended [14,46,59]. This fact should also be taken into account that the nature of these problems always dictates a compromise between designing a realistic and efficient solution algorithm, and considering as many as possible characteristics of the tracking systems described above. For instance, Coleman et al. [14] considered a minimizing tracking error problem with holding a small number of assets in their designed portfolio and considering uncertainty associated with the future value of the securities. They
proposed a graduated non-convexity method in which the discontinuous cardinality constraint function was replaced with a sequence of continuous approximation. It was pointed out that the efficiency of this method depends on the sequence of the approximation and how it approaches the original form of tracking error minimization problem. In another work performed by Gaivoronski et al. [45], in addition to the aforementioned characteristics considered by Coleman et al. [14], a rebalancing strategy for making any decision in future was considered. Different concepts of benchmark tracking models both static and dynamic were examined to determine the best relative strategy to satisfy desired requirements of different investors. It was concluded that the performance of the tracking portfolios is a function of the investment time horizon. The performance of this type of portfolios generally enhance with increasing the length of the optimizing period, number of assets in the portfolio as well as rebalancing. Stoyan et al. [58] considered the exposure of the tracking portfolio and the index to the similar economic sectors and the restriction on the transaction cost in addition to the characteristics considered by last two mentioned studies in their proposed index tracking optimization model. They used their developed algorithm to solve a two-stage stochastic mixed integer programming. In their developed strategy unlike other stochastic decomposition algorithm in which decomposition is performed on the time stages or scenario generation, the decomposition was done through two constraints considered in the model. It was pointed out that applying non-anticipativity between decomposition might results in computational challenges for the solution process as well as CPU time, thus non-anticipativity constraints were implicitly added to the sub-problems. In their proposed approach the main problem decomposed into sub-problems through sector constraints which accompanied by a relaxation. They defined different sectors and the sector exposure was determined through defining an indicator. This developed strategy showed competitive tracking values. However, they pointed out that the trade-off on their solution approach resulted from the number of names to hold constraint.

An index tracking error problem with constraints on its number of total assets can mathematically be formulated as a minimization problem. The objective function of this problem is defined to measure the tracking error subjected to a discrete counting
function for considering those assets in which investment is made as well as any other possible linear simple constraints depending to the problem. As a measure of how closely a portfolio typically trails an index, a tracking error can be defined as a difference between the portfolio and index values. Therefore, a basic index tracking portfolio with constraint on total assets can be formulated as follows [14, 44]:

\[
\min \sum_{i=1}^{n} T(x_i) \tag{4.10}
\]

s.t. \[
\sum_{i=1}^{n} P(x_i) \leq K \tag{4.11}
\]

\[
\sum_{i=1}^{n} x_i = 1 \tag{4.12}
\]

\[
x_i \geq 0 \quad \forall \ i = 1, ..., n. \tag{4.13}
\]

where \(x_i\) is the percentage weight of asset \(i\) selected in the portfolio, \(T(x)\) is the tracking error function, \(P(x_i) = 1\) if \(x_i \neq 1\), otherwise \(P(x_i) = 0\), and \(K\) is the upper bound on the number of the securities held in the portfolio. The related constraint on the number of assets in the portfolio, \(\sum_{i=1}^{n} P(x_i) \leq K\), can also imply the restriction imposed upon the management cost. As pointed out, this is a NP-hard problem and to solve this problem the imposed constraints are relaxed, and to find an optimum solution most developed algorithms have a heuristic nature.

Different models for international index funds and corporate bond index funds are described by Zenios [2] as well. Coleman et al. [14] reviewed few different ways to define the objective function or measuring the tracking error for the index tracking error minimization problem. For instance, Beasley et al. [46] measured the tracking error considering the historical prices of the stock and index as:

\[
T(x_i) = \frac{1}{\tau} \left( \sum_{t=1}^{\tau} |r_t(y) - R_t|^p \right)^{1/p} \tag{4.14}
\]
Where $x_i$ here is the number of units of stock $i, i = 1, \ldots, n$ in the index tracking portfolio, $\tau$ is the number of time periods in which the historical values for both stock and index are observed, and $\rho > 0$ is a parameter for penalising. $R_t$ is the index return for the time period $t - 1$ and $t$, and $r_t(y)$ is the single period time return obtained from the new index tracking portfolio which can be defined respectively as follows:

$$R_t = \ln \frac{I_t}{I_{t-1}}$$  \hspace{1cm} (4.15)

$$r_t = \ln \frac{p_t^r x}{p_{t-1}^r x}$$  \hspace{1cm} (4.16)

where $I_t$ is the index price and $P$ is also the index price. It should be noted that the tracking error function is not convex. Another example for the common objective function employed for tracking error measurement is a function by which it is possible to quantify the active risk of a tracking portfolio based on the covariance matrix of the stock return. This is the method was used by Jansen et al. [60]. Considering a tracking error portfolio with portfolio weights vector $y$ with respect to an index with weights vector $w$, and covariance matrix $Q$ of the stock returns, the tracking error function can be expressed as:

$$T(y) = (y - w)Q(y - w)$$  \hspace{1cm} (4.17)

Obviously, this expression is a quadratic function and convex, and consequently mathematically convenient to deal with. More detailed discussion on this tracking error function is provided in the literature [14, 60].

### 4.3 Stochastic programming for index tracking

As it was described in the last chapter for stochastic dynamic portfolio optimization strategies, for the dynamic index tracking problem as briefly described in this chapter, the model should be developed considering different possible scenarios at each multiple trading dates. At each time step, a set of potential transactions for all securities in the
index along with the index composition are available. Contrary to the basic models introduced in the previous section, for the dynamic modeling, it is possible to rebalance the index tracking portfolios at each time stage according to the available information at that particular stage. This is a more realistic approach since it is possible to update the model according to the incoming information, and consequently to have a model which is able to track a benchmark as close as possible in the course of the investment time horizon. Therefore, this type of models is designed based on a multistage scenario tree developed according to the specifications of the investment and the strategy intended to follow for managing the designed portfolio. For this purpose, both securities and index returns are labelled by each related time and state, the same as performed for the stochastic dynamic optimization models. At each time step the model will then be rebalanced based on the new information arriving from the last to the current time period throughout the whole investment time horizon, $t = 0, \ldots, T$.

In this section, a stochastic programming for index tracking introduced by Zenios [2] is briefly reviewed. For this purpose, the problem variables and parameters representing buying and selling securities, risk free asset investment, and securities held in the indexed portfolio should properly be defined. Face value of initial holding of security $i$ and initial holdings in the risk free asset are denoted by $b_{0i}$ and $d_0$, respectively. Face values invested in and sold of security $i$ at time period $t$ in the state $s \in \Omega$ are denoted by $x_{it}^s$ and $y_{it}^s$, respectively. Position of security $i$ at time period $t$ in the state $s \in \Omega$ is denoted by $h_{it}^s$. Cash investment in the risk free rate at time period $t$ in the state $s \in \Omega$ is denoted by $d_{it}^s$. Amortization factor for security $i$ at time period $t$ in the state $s \in \Omega$ to the next time period $t + 1$ in the state $s + 1 \in \Omega$ is denoted by $\theta_{it}^{s+1}$. Ask and bid prices of security $i$ at time period $t$ in the state $s \in \Omega$ are denoted by $A_{it}^s$ and $B_{it}^s$, respectively. Cash generated by security $i$ from time period $t$ to the next time period $t + 1$ is denoted by $D_{it}^{s+1}$. Finally, rate of return for the risk free asset between stages $t$ and $t + 1$ is denoted by $r_t^{s+1}$.

To model the problem there are two sets of constraints, the first set concerns with the cash flow for the risk free asset and the second set of constraints are related to each
security in the tracking portfolio for each stage and states. Since for the first stage all prices and the composition of the portfolio are known, the portfolio value is therefore defined as:

\[ D_0 = d_0 + \sum_{i=1}^{n} B_{i0}^0 b_{i0} \]  \hspace{1cm} (4.18)

The following is the inventory balance constraint for the first stage:

\[ h_{i0}^0 = d_o + x_{i0}^0 - y_{i0}^0 \]  \hspace{1cm} (4.19)

The cash flow balance is also defined as:

\[ \sum_{i=1}^{n} B_{i0}^0 y_{i0}^0 + d_o = \sum_{i=1}^{n} A_{i0}^0 x_{i0}^0 + v_0^{s+1} \]  \hspace{1cm} (4.20)

For the constraint concerning the recourse decision, \( 0 < t < T \), all decisions should be conditioned on their related state considering all information and data available from the previous stage. The asset inventory balance as did for the first stage imposes a constraint on the position of each security, sold or remained in the portfolio. It should be equal to the amount of face value carried from previous stages and all purchases at the current stage.

\[ h_{it}^s = h_{i(t-1)}^{s-1} + x_{it}^s - y_{it}^0, \text{ for all } i = 1, \ldots, n \text{ and } s \in \Omega. \]  \hspace{1cm} (4.21)

However, the asset inventory constraint can be written in a more comprehensive format to include the amortization effects resulted from instruments in the portfolio such as options or mortgage securities which might be prepaid:

\[ h_{it}^s = a_{i(t-1)}^s h_{i(t-1)}^{s-1} + x_{it}^s - y_{it}^0, \text{ for all } i = 1, \ldots, n \text{ and } s \in \Omega. \]  \hspace{1cm} (4.22)

The next constraint states that the total investment in the new securities and the risk free asset should be equal to the added value of the portfolio through its existing
constituents, and also cash added from sales and reinvestment at the previous time period at predecessor state. Therefore, the cash flow constraints for stages $0 < t$ are:

$$\sum_{i=1}^{n} D_{i(t-1)}^{s} h_{i(t-1)}^{s-1} + \sum_{i=1}^{n} B_{it}^{s} y_{it}^{s} + (1 + r_{it}^{s}) d_{t-1}^{s+1} = \sum_{i=1}^{n} A_{it}^{s} x_{it}^{s} + v_{t}^{s+1}, \quad (4.23)$$

for all $i = 1, ..., n$ and $s \in \Omega$.

Finally, at the ending point of the investment time horizon the portfolio value should be calculated and consequently compared with the index value for all states. Therefore, for the corresponding constraint, the portfolio value, which is the total amount of different assets as well as cash for each state, and the value $D_0$ invested in the index, can be determined as:

$$D_{T}^{s} = d_{T}^{s} + \sum_{i=1}^{n} B_{iT}^{0} h_{iT}^{s} \quad (4.24)$$

$$D_{T}^{s} = V_0 (1 + R_{T}(w; r_{T}^{s})) \quad (4.25)$$

where $r_{T}^{s}$, which is the rate of return of security $i$, and the portfolio rate of return can respectively be defined as:

$$r_{T}^{s} = \frac{B_{iT}^{s} - B_{0i}^{0}}{B_{0i}^{0}} \bigg|_{i=1}^{n} \quad (4.26)$$

$$R_{p}^{s} = \frac{V_{T}^{s} - V_0}{V_0} \quad (4.27)$$

Considering all constraints, now one can build a model to maximize the expected return of the portfolio while the designed portfolio can closely track the index value.

$$\max \sum_{s \in \mathcal{S}} p_s R_{p}^{s} \quad (4.28)$$

$$s.t. \quad h_{i0}^{0} = d_{o} + x_{i0}^{0} - y_{i0}^{0}, \quad \text{for all } i = 1, ..., n \quad (4.29)$$
\[ \sum_{i=1}^{n} B_{i0}^0 y_{i0}^0 + d_o = \sum_{i=1}^{n} A_{i0}^0 x_{i0}^0 + v_0^{s+1} \]  

(4.30)

\[ h_{it}^s = h_{i(t-1)}^{s-1} + x_{it}^s - y_{it}^0, \text{ for all } i = 1, \ldots, n; t = 0, \ldots, T \text{ and } s \in \Omega. \]  

(4.31)

\[ \sum_{i=1}^{n} D_{i(t-1)}^s h_{i(t-1)}^{s-1} + \sum_{i=1}^{n} B_{it}^s y_{it}^s + (1 + r_{it}^s) d_{t-l}^{s+1} = \sum_{i=1}^{n} A_{it}^s x_{it}^s + v_{l}^{s+1}, \]  

(4.32)

for all \( i = 1, \ldots, n; t = 0, \ldots, T \) and \( s \in \Omega. \)

\[ -\varepsilon < V_{l}^{s} - V_{i}^{s} < \varepsilon, \text{ for all } s \in \Omega. \]  

(4.33)

\[ x, y, h \geq 0. \]  

(4.34)

As seen in the model, the constraints defined for the index tracking to restrict the difference between the portfolio and the index values between predefined small numbers \(-\varepsilon\) and \(\varepsilon\), were only imposed on the end of the investment time horizon. Therefore, there was no guarantee that the portfolio tracks the index during the investment time periods. However, it was pointed out that the presented model can be employed for tracking the index throughout the investment time horizon by imposing these constraints for all time periods, \( t = 0, \ldots, T \).

### 4.4 A Stochastic network flow mixed-integer programming approach for the index tracking problem

In this section, the stochastic network flow mixed-integer programming approach used in Section 3.4 for the financial planning is adapted for the index tracking problems. The problem formation in this section is presented for the two stage network; however this approach can be extended for benchmark tracking problems with multistage network structures. As described in the preceding sections in this chapter, an index tracking problem is a passive portfolio management strategy by which a portfolio holding relatively a small number of securities tracks the performance of a benchmark. This approach assists portfolio managers not only to assess the performance of their portfolio...
with a target value, but also to design a portfolio with a restricted transaction as well as administration costs. Considering the index tracking formulation criteria presented in this chapter, the objective function for the tracking problem designed based on the network flow approach can be written as follows:

\[
\min_{x,p} \sum_{(i,j) \in A^0, i \neq j} T_{ij}^{00} x_{ij}^{00} + \sum_{s \in S} P(s) \left( \left\| \sum_{(i,j) \in A^1(s)} (1 - T_{ij}^{11}(s)) x_{ij}^{11}(s) - l_i^1(s) \right\| \right)
\]

(4.35)

where \( l_i^1(s) \in \mathbb{R} \) is the index value at the second stage. All other notations are kept the same as presented in Section 3.4. In the objective function \( T_{ij}^{00} \) are known while \( T_{ij}^{11} \) and \( l_i^1(s) \) are unknown parameters.

The objective function is designed to force the net portfolio value at the first stage to be equal or relatively as close as possible to the target value obtained for the second stage. As the objective function explicitly shows, the first term is to keep the transaction cost minimum during the asset allocation, and the second term is the stochastic stage term. The initial investment is scaled accordingly to the known net target value \( l^0 \) at the first stage as \( b_i^0 = l^0/K \). Consequently, the constraints for this problem can be defined as:

\[
\sum_{(i,j) \in A^0} x_{ij}^{00} - \sum_{(j,i) \in A^0, j \neq i} x_{ji}^{00} \leq b_i^0 \quad \forall \ i \in N,
\]

(4.36)

\[
\sum_{(i,j) \in A^0} x_{ij}^1(s) - \sum_{(j,i) \in A^0} (1 - T_{ij}^1(s)) x_{ji}^1(s) \leq R_{ij}^{01}(s) x_{ji}^{11}(s) \quad \forall \ i \in N, s \in S,
\]

(4.37)

\[
\sum_{i \in N} y_i^{00} = K, \quad y_i \in \{0,1\}
\]

(4.38)

\[
l_{ij}^{00}(y_i^{00}) \leq x_{ij}^{00} \leq u_{ij}^{00}(y_i^{00}) \quad \forall \ i \in N,
\]

(4.39)

\[
l_{ij}^{11}(y_i^{11}) \leq x_{ij}^1(s) \leq u_{ij}^{11}(s)(y_i^{11}) \quad \forall \ i \in N, (i,j) \in A^1(s), s \in S.
\]

(4.40)
The first two set of constraints are the network flow conservation constraints. As seen before, these two set of constraints demonstrate the balance among the input and output flows on all nodes at the first and second time stages. The third constraint is the cardinality constraint and the last two sets of constrains impose the lower and upper bounds on the problem design variables. Since in this modelling approach the design variables are set to be real numbers, as pointed out by Stoyan et al. [58] this may result in errors due to the round-offs of fractional security investments. Coleman et al. [14] developed their index tracking problem by presenting the number of units of different securities, $z_i$ in the portfolio $z$ based on the percentage holding of each securities $x_i$ as:

$$x_i = \frac{p_i^t z_i}{(P^t)^T z}$$  \hspace{1cm} (4.41)

where $P_i^t$ is the price of security $i$ at time $t$ and $P^t$ is the vector of security prices at time $t$. Then, the tracking errors were measured as functions of either $x$ or $z$. However, by considering a well-defined round-offs algorithm, the associated errors were restricted within a specified range without compromising the results obtained from the tracking problem [61]. More details on the round-off effect of integer variables on the results of an optimization process can be found in the literature [62]. To be able to solve the stochastic network flow mixed-integer programming model for the index tracking formulated above, the method presented by Konno et al. [63] is used. In their method, they defined an equivalent linear form of the MVO model under multivariate normal return distributions which was called Mean-Absolute Deviation is used. Using this method, the index tracking model defined based on the network flow structure is equivalent to the following model:

$$\min_{x, p} \sum_{(i,j) \in A^0, i \neq j} T_{ij}^{00} x_{ij}^{00} + \sum_{s \in S} P(s) \left( q_{1+}^0 (s) + q_{0-}^0 (s) \right)$$ \hspace{1cm} (4.42)

s.t. \hspace{1cm}

$$\sum_{(i,j) \in A^0} x_{ij}^{00} - \sum_{(j,i) \in A^0, j \neq i} x_{ji}^{00} \leq b_i^0 \hspace{1cm} \forall i \in N,$$ \hspace{1cm} (4.43)

$$\sum_{(i,j) \in A^{it}(s)} x_{ij}^1 (s) - \sum_{(j,i) \in A^{is}(s)} (1 - T_{ji}^1 (s)) x_{ji}^1 (s) \leq R_i^{01} (s) x_{ii}^{01} (s) \hspace{1cm} \forall i \in N, s \in S,$$ \hspace{1cm} (4.44)
\[
\sum_{i \in N} y_{i}^{00} = K, \quad y_i \in \{0,1\} \tag{4.45}
\]

\[
l_{ii}^{00}(y_{i}^{00}) \leq x_{ii}^{00} \leq u_{ii}^{00}(y_{i}^{00}) \quad \forall \ i \in N, \tag{4.46}
\]

\[
l_{ij}^{11}(s)(y_{i}^{11}) \leq x_{ij}^{11}(s) \leq u_{ij}^{11}(s)(y_{i}^{11}) \quad \forall \ i \in N, (i,j) \in A^{11}(s), s \in S, \tag{4.47}
\]

\[
\sum_{(i,j) \in A^{s}} \left(1 - T_{ij}^{11}(s)\right)x_{ij}^{11}(s) - l^{1}(s) = q^{1+}(s) - q^{0-}(s) \quad \forall \ s \in S, \tag{4.48}
\]

\[
q^{1+}(s), q^{0-}(s) \geq 0 \quad \forall \ s \in S. \tag{4.49}
\]

where the new set of constraints as well as new variables \(q^{1+}(s)\), and \(q^{0-}(s)\) allow for the linear transformation of the former network flow index tracking problem with the absolute value terms in the objective function.

### 4.5 Numerical results and analysis

In this section, the same as presented in Section 3.5 different aspects of the developed model for the index tracking problem is discussed through numerical experiments. Mathematical model is solved for the two-stage portfolio optimization process using the same method and computer described in 3.5. In the first case study, it is planned to design a portfolio with three securities, therefore, \(K\) is equal to three from a market with total four securities, \(n = 4\). Four different scenarios are generated, \(s = 4\). Table 4.1 and Table 4.2 list the return values and transaction costs for all scenarios, respectively.

**Table 4.1 Return values for four scenarios**

<table>
<thead>
<tr>
<th>(R_i(s))</th>
<th>(n = 1)</th>
<th>(n = 2)</th>
<th>(n = 3)</th>
<th>(n = 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S=1</td>
<td>1.9222</td>
<td>1.9346</td>
<td>1.5452</td>
<td>1.1127</td>
</tr>
<tr>
<td>S=2</td>
<td>1.7434</td>
<td>1.5377</td>
<td>1.3241</td>
<td>0.1184</td>
</tr>
<tr>
<td>S=3</td>
<td>1.2584</td>
<td>1.1869</td>
<td>0.7028</td>
<td>0.6939</td>
</tr>
<tr>
<td>S=4</td>
<td>0.9962</td>
<td>0.8928</td>
<td>0.7305</td>
<td>0.6095</td>
</tr>
</tbody>
</table>
Table 4.2 Transaction cost for the second stage with four scenarios

<table>
<thead>
<tr>
<th>First scenario</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ij}^{(1)}$</td>
<td>$j = 1$</td>
<td>$j = 2$</td>
<td>$j = 3$</td>
<td>$j = 4$</td>
</tr>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0799</td>
<td>0.0823</td>
<td>0.0758</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0797</td>
<td>0</td>
<td>0.0808</td>
<td>0.0812</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0830</td>
<td>0.0807</td>
<td>0</td>
<td>0.0816</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0772</td>
<td>0.0834</td>
<td>0.0816</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Second scenario</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ij}^{(2)}$</td>
<td>$j = 1$</td>
<td>$j = 2$</td>
<td>$j = 3$</td>
<td>$j = 4$</td>
</tr>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0842</td>
<td>0.0836</td>
<td>0.0805</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0848</td>
<td>0</td>
<td>0.0798</td>
<td>0.0812</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0826</td>
<td>0.0751</td>
<td>0</td>
<td>0.0753</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0808</td>
<td>0.0762</td>
<td>0.0770</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Third scenario</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ij}^{(3)}$</td>
<td>$j = 1$</td>
<td>$j = 2$</td>
<td>$j = 3$</td>
<td>$j = 4$</td>
</tr>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0462</td>
<td>0.0454</td>
<td>0.0519</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0454</td>
<td>0</td>
<td>0.0513</td>
<td>0.0499</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0498</td>
<td>0.0464</td>
<td>0</td>
<td>0.0503</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0469</td>
<td>0.0468</td>
<td>0.0503</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fourth scenario</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ij}^{(4)}$</td>
<td>$j = 1$</td>
<td>$j = 2$</td>
<td>$j = 3$</td>
<td>$j = 4$</td>
</tr>
<tr>
<td>$i = 1$</td>
<td>0</td>
<td>0.0177</td>
<td>0.0191</td>
<td>0.0160</td>
</tr>
<tr>
<td>$i = 2$</td>
<td>0.0199</td>
<td>0</td>
<td>0.0170</td>
<td>0.0164</td>
</tr>
<tr>
<td>$i = 3$</td>
<td>0.0235</td>
<td>0.0206</td>
<td>0</td>
<td>0.0166</td>
</tr>
<tr>
<td>$i = 4$</td>
<td>0.0237</td>
<td>0.0214</td>
<td>0.0158</td>
<td>0</td>
</tr>
</tbody>
</table>
Transaction costs for the first stage are considered the same as before, \( T_{ij}^{00} = 0.05 \), as listed in Table 3.2. The initial values for the four available securities are considered as:

\[
\begin{align*}
n_1 &= 107.37, & n_2 &= 55.21, & n_3 &= 143.12, & n_4 &= 122.86
\end{align*}
\]

and therefore, the initial investment for each securities, \( b_i^0 \), is equal to 107.14. Ultimately, the index value, \( I^1(s) \), for the scenarios are:

\[
I^1(1) = 671.05, \quad I^1(2) = 476.14, \quad I^1(3) = 386.49, \quad I^1(4) = 335.69
\]

Considering the aforementioned input deterministic and stochastic parameters, by solving the model, final results for the first stage variables, \( x_{ij}^{00} \), and second stage variable, \( x_{ij}^{11}(s) \) for \( s = 1, \ldots, 4 \), respectively, are as follows:

\[
x^{00} = \begin{bmatrix} 309.08 & 0 & 0 & 0 \\ 0 & 107.14 & 0 & 0 \\ 94.79 & 0 & 12.34 & 0 \\ 107.14 & 0 & 0 & 0 \end{bmatrix}, \quad x^{11}(1) = \begin{bmatrix} 577.82 & 100 & 100 & 0 \\ 100 & 191.20 & 100 & 0 \\ 100 & 100 & 2.75 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\]

\[
x^{11}(2) = \begin{bmatrix} 522.09 & 100 & 100 & 0 \\ 100 & 148.80 & 100 & 0 \\ 100 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad x^{11}(3) = \begin{bmatrix} 378.45 & 100 & 100 & 0 \\ 100 & 117.89 & 100 & 0 \\ 98.99 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\]

\[
x^{11}(4) = \begin{bmatrix} 303.56 & 100 & 100 & 0 \\ 100 & 91.82 & 100 & 0 \\ 100 & 100 & 5.39 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\]

In addition, Figure 4.1 schematically shows the results presented for the two-stage network with four security nodes and four scenarios.
Figure 4.1 Results for the three scenarios of a portfolio consisting of three securities
As seen in Figure 4.1 as well as the numerical results, the designed portfolio consists of the first, second and third securities denoted by node 1 to 3 as selected in the first stage. In the second stage or portfolio rebalancing stage, the resultant portfolios of the second and third scenarios only consist of the first and second securities; while the portfolios resulted from the first and fourth scenarios still consist of all three securities invested in during the asset allocation process. The final value of the objective function is the sum of each portfolio value resulted from the four generated scenarios, considering their associated probabilities. In this case study, the probability associated with the first and fourth scenarios, the two extreme events relative to the others, is considered 20 percent. For the other two scenarios, the probability associated with their occurrence is assumed to be 30 percent. Figure 4.2 shows the tracking errors for each scenario and the whole portfolio. The values of the tracking errors shown in this figure are a reflection of the characteristics of the problem solved. For this particular problem to track the index there are three choices out of four options.

![Figure 4.2 Tracking error for the portfolio and each scenario](image)

To explicitly demonstrate the role of minimizing the transaction costs in the first stage, the model is solved once with the same structure as described before for the first case study. Another model was built and solved with the same input data but without
minimizing its corresponding first stage transaction costs. Their results were then compared as shown in Figure 4.3 for the first and second model with and without minimizing the transaction costs in the first stage. As clearly seen in the figure, the two models have a totally different capital distribution among their selected securities. In addition, regarding the cardinality constraint, both designed portfolios have the same number of securities but different numbers of transactions. This shows the importance of minimizing the transaction costs parallel to the cardinality constraint in order to have a full control on the portfolio management costs. It is evident that the increased number of transactions for the real portfolios with a considerable number of securities definitely affects their final added value.

![Figure 4.3 Asset allocation: a) with and b) without minimizing the transaction costs](image)

The developed model has $n^2$ continuous and $n$ binary variables at the first stage, and $n^2 \times s + 2s$ continuous variables at the second stage ($n$ and $s$ are the number of nodes and scenarios, respectively). Therefore, the total continuous and binary variables is $n^2 (1 \times s) + 2s + n$. Number of variables versus the number of securities, $n = 1, \ldots, 500,$
and generated scenarios, \( s = 1, \ldots, 40 \), is shown in Figure 4.4. Accordingly, the total number of constraints for this model is \( 3 \times n + n \times s(1 + 2s) + s + 1 \). Number of constrains versus the number of scenarios and securities for \( s = 1, \ldots, 40 \) and \( n = 1, \ldots, 500 \) is shown in Figure 4.5. Therefore, for the first case study presented in this section, there were 92 continuous and binary variables and the total of 161 constraints. Figure 4.6 and Figure 4.7 show the running time and gap versus different number of securities (i.e., cardinality constraint) of a portfolio designed from total number of 20 securities.

As seen, running time and gap increase with the number of securities held in a portfolio with visible increasing rates. Considering the number of variables and constraints of the real problems as well as their corresponding running time and gap obtained from the commercial software, as pointed out in the last chapter, to have an efficient solution process, designing a a problem specific algorithm is inevitable. This should be done to take advantage of many desired prominent features of the developed model based on the network flow structure.

![Figure 4.4](image)

**Figure 4.4** Number of Variables corresponding to the numbers of securities and scenarios
Figure 4.5 Number of Constraints corresponding to the numbers of securities and scenarios

Figure 4.6 Running time versus number of securities in the portfolio for n=20
Figure 4.7 Gap versus number of securities in the portfolio for n=20
Chapter 5

Conclusion and future research

In this research, a modelling approach was presented for the portfolio design and management. The main goal was to develop a model that can realistically capture the main goals and requirement of an investment portfolio such as restriction on the transaction cost or on the number of trades. In addition, one of the main features considered in the modelling development process in this work was the capability of rebalancing the designed portfolio by the portfolio managers throughout the investment time horizon whenever new or more information are available. Therefore, the proposed model is a scenario based model and built for multi-period dynamic portfolio optimization. For this purpose, the network flow structure was used to develop this financial model. The network flow structure has been used to model many novel and important planning problems in different fields. The network flow structure for financial applications can highly facilitate portfolio representation as well as revisions. Particularly, it is very efficient for the large-scale planning problems. The proposed modelling approach as pointed out is developed for the multistage financial optimization process. However, at large it can be divided into two main stages, portfolio asset allocation and portfolio management.

As described in the second and third chapters, portfolio optimization models with a cardinality constraints as a part of their decision support system is considered NP-hard and therefore many available models have not considered this constraint directly in their modelling structure, for instance, the models proposed for the investment planning using
the network flow structures [37,42]. As described, the cardinality constrained can be considered as the upper bond imposing on the administration costs. Design a model having this type of constraints requires the presence of the binary variables in its mathematical formulation. Therefore, the corresponding optimization problem is not considered a quadratic program anymore. The resultant model is a quadratic mixed integer program, and consequently as mentioned it is considered as a NP-hard problem. Regardless of many advantages that the proposed multistage stochastic mixed-integer programming approach for financial planning with network flow structure offer, the main advantage of this model compared to the models developed using network flow structure is considering the cardinality constraint in the model. In general, solution of a model may result in a portfolio with a large number of securities that some of them might be in the portfolio with only small positions. This is not a desirable situation, as a portfolio consisting of a large number of securities implies a larger total transaction cost for each time that the portfolio requires to be rebalanced. To solve this shortcoming and at the same time to have a well-diversified portfolio, in some approaches transaction costs are considered as a term in the objective function and are minimized. Although in this case there is a restriction imposed on the number of the securities in the portfolio, there is no direct control on their exact number. In the proposed approach presented in this research, as described in more detail in Chapter 3, the number of the securities is explicitly controlled by a cardinality constraint. This constrain imposes a limitation on the transaction costs indirectly, however, in order to have the transaction costs fully under the control, the objective function of the model is designed to minimize the total transaction costs while maximizing the total value of the investment. This is considered to restrict the flows among the connecting arcs of the selected securities included in the portfolio during the asset allocation step.

In addition to the cardinality constraints, two other main constraints which are common for all network structures are the flow conservation constraints. For this problem, there were two types of conservation constraints; the first type was designed for the first step which is asset allocation process. The second type imposed the
conservation constraints on all nodes of the flow network structure for the second and other successive rebalancing stages.

The same approach is used to develop another optimization model for an interesting financial management strategy which is known as the index tracking problems. Solving these problems and using this strategy, portfolio managers can evaluate the performance of their portfolios by comparing it with a target value as a benchmark. This is a passive strategy which is not intended to outperform the market performance. All technical complexities described for the first developed model apply to this type of problems as well. Therefore, all advantageous features listed for the network structures can be utilized to facilitate a more realistically practical mathematical formulation and solution process of this type of problems.

In the last sections of Chapter 3 and Chapter 4, experimental results for a two-stage model were presented to show different aspects of the proposed approaches. It was shown how considering the cardinality constraints directly into account can improve the realistic aspects of the main elements which are critical to the framework of a portfolio management process. In conclusion, the main outcomes and contributions of this research can be summarized as follows:

- Developing a modelling approach built on the network flow framework to facilitate capturing the main elements of a realistic dynamic portfolio optimization as discussed in Chapter 3 and Chapter 4.
- Considering a set of realistic financial portfolio managing constraints specifically the cardinality constrains. Cardinality constrain although makes the model realistic, it significantly increases the level of complexity associated with the solution process.
- Presenting the initial solutions of the models to show the main prominent advantages of the developed portfolio optimization models including their solvability and practicability.
5.1 Future research

The topics presented in this thesis have mainly focused on the main aspects of the portfolio modelling process using the network flow structures. To fully exploit different advantageous characteristics of the proposed approaches, there are several other features of the developed models which warrant more detailed investigation as well as development. Therefore, considering the work presented in this thesis and its several promising features, the following interesting research avenues recommended for the future work on this topic.

The models, presented in Chapter 3 and Chapter 4, were developed to demonstrate the basic capabilities of the proposed portfolio modelling approach. It would be interesting to consider some more detained aspects for financial planning such as borrowing or short sales into account. To fully study and investigate the multistage performance of the model, the proposed multistage mathematical model can be solved for more than two stages. The arc multipliers or input parameters mainly return values can be prepared using real data. For this purpose, methods for scenario generations described in Chapter 2 can be used along with the available data of the financial markets. An optimization model, for instance the one proposed by Hoyland et al. [28] can be used prior to the main financial optimization model to generate desired scenarios. An interesting and promising research topic can be integration of the stochastic scenario generation and multistage financial planning. Another aspect that can further be investigated is that different capacities can be assigned for different arcs in the model. Furthermore, it is possible to consider different types of securities with different policies. However, the main and most interesting future research is to develop solution algorithm to solve the model such heuristic strategies explained in Chapter 3. Theses parallel solution strategies should be designed in order to increase the accuracy of the solution as well as to decrease solution time for the real size problems. These aspects should be compared with those of the commercial solvers. The implementation issues of these strategies should comprehensively be investigated as well.


27. D. Pachamanova, F. Fabozzi, Simulation and Optimization in Finance: Modeling with MATLAB, @Risk, or VBA. Wiley, 2010.


