UNIVERSITÉ DE MONTRÉAL

A PORTFOLIO OPTIMIZATION MODEL

HASSAN RAHNAMA

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L’OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INDUSTRIEL)

DÉCEMBRE 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

A PORTFOLIO OPTIMIZATION MODEL

présenté par : RAHNAMA Hassan

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. ROUSSEAU Louis-Martin, Ph. D., président

M. SAVARD Gilles, Ph. D., membre et directeur de recherche

M. AUDET Charles, Ph. D., membre

DEDICATION

Dedicated to my father

ACKNOWLEDGEMENTS

I would like to cordially appreciate my family for their ever-growing encouragement, motivation, and support in all the time.

I would also like to express my sincere gratitude to my research advisor Prof. Gilles Savard for his patience, support, and guidance throughout my studies.

A very special thanks to my friends and all those who helped me in any way.

RÉSUMÉ

Ce mémoire étudie le problème d’optimisation de portefeuille moyenne-variance (MV) de Markowitz avec contrainte de cardinalité et bornes sur les variables. C’est un problème NP-Difficile modélisé à l’aide d’un programme MIQP. La performance du portefeuille MV optimal généré est évaluée à l’aide de la méthode exacte *Branch-and-Bound* (BB) qui fournit une solution optimale globale en comparaison avec les méthodes les plus performantes de la littérature, comme l'écart absolu moyen, la différence moyenne de Gini et la valeur conditionnelle à risque. Ces méthodes alternatives utilisent différentes mesures de risque. De plus, nous avons appliqué pour la première fois une approximation externe (*Outer Approximation* - OA) au problème MV et nous avons proposé une nouvelle heuristique de branchement afin de faire face à la difficulté de résolution des grandes instances du problème. Ces dernières approches utilisent des techniques de décomposition. La classification des résultats numériques montre que la méthode exacte (BB) est efficace pour les problèmes de petite taille, que la méthode OA surpasse les autres méthodes pour les instances de taille moyenne et que l’heuristique proposée de branchement est efficace pour les instances de taille importante où les méthodes BB et OA ne sont pas applicables. À cause de la complexité de la structure du problème, les méthodes exactes sont incapables de résoudre les problèmes de taille importante dans un temps raisonnable. C’est pourquoi les méthodes heuristiques sont développées pour faire un compromis entre la précision de la solution et le temps de résolution.

ABSTRACT

This thesis investigates the Markowitz' Mean-Variance (MV) portfolio optimization model with cardinality constraint and bounds on variables which is MIQP model and known as an NP-Hard problem. We evaluate the performance of the optimal MV-portfolio generated by Branch-and-Bound (BB) algorithm as an exact method which provides a global optimal solution in comparison with the most effective alternative methods in literature such as Mean Absolute Deviation, Gini Mean Difference and Conditional Value at Risk. These alternative methods make use of different risk measures. In addition, we applied an Outer Approximation (OA) algorithm for the MV problem for the first time as well as proposing a new Heuristic Branching algorithm to deal with the difficulty of the problem for large instances. The later approaches utilize some sort of problem decomposition. With the classification of the numerical results, we showed that the exact method (BB) is efficient for small size problem while for medium size problem the OA outperforms the other methods and the proposed Heuristic Branching algorithm is efficient for large size problems since BB and OA are not applicable in this category. Due to the complexity of the problem structure, exact methods are not capable of solving large size problem in a reasonable time budget. Thus, heuristic methods are developed to trade-off between the precision of the solution and computational time.

TABLE OF CONTENTS

DEDICATION ……………………………………………………………………………….. III

ACKNOWLEDGEMENTS …………………………………………………………………….. IV

RÉSUMÉ …………………………………………………………………………………..... V

ABSTRACT ……………………………………………………………………………….. VI

TABLE OF CONTENTS ……………………………………………………………………. VII

LIST OF TABLES …………………………………………………………………….... IX

LIST OF FIGURES ……………………………………………………………………..... X

LIST OF SYMBOLS AND ABBREVIATIONS ……………………………………………….. XI

CHAPTER 1 INTRODUCTION ……………………………………………………………... 1

1.1. Context …………………………………………………………………………………. 1

1.2. Organization of the Thesis …………………………………………………………...... 2

1.3. Contribution ………………………………………………………………………..... 3

1.4. Financial Market …………………………………………………………………….. 4

1.5. Risk and Return …………………………………………………………………….…. 4

1.6. Portfolio Diversification ……………………………………………………………..... 4

1.7. Capital Asset Pricing Model ………………………………………………………… 8

CHAPTER 2 LITERATURE REVIEW ………………………………………….……. 10

2.1. Portfolio Optimization Problem ……………………………………………….… 10

2.2. Portfolio Optimization with Real Features ………………………………….…… 12

2.2.1. Cardinality Constraints ……………………………………………….…... 12

2.2.2. Bounds on investment …………………………………………………….….... 13

2.3. Problematic ……………………………………………………………………….… 13

2.4. Mean Absolute Deviation (MAD) …………………………………………….… 15

2.5. Gini Mean Difference (GMD) ………………………………………………….... 16

2.6. Value at Risk (VaR) …………………………………………………………..... 16

2.6.1. Minimizing CVaR …………………………………………………... 17

2.7. Mixed-Integer Nonlinear Programming Models (MINLP) ……………………..... 19

2.8. Fundamental Methods to Solve MINLP ……………………………………..… 20

2.8.1. Piecewise Linearization ………………………………………………... 22

2.8.2. Lower Bounds ……………………………………………………….… 22

2.8.3. Upper Bounds ……………………………………………………….…. 24

2.8.4. Valid Cuts ……………….…………………………………………….... 25

2.9. Outer Approximation Algorithm …………………………………………….…... 28

CHAPTER 3 METHODOLOGY AND MODELS ……………………………….…... 31

3.1. Outer Approximation Algorithm ……………………………………………… 32

3.2. Proposed Heuristic Branching Algorithm ………………………………………... 35

CHAPTER 4 NUMERICAL RESULTS AND IMPLEMENTATION ……………...…. 40

4.1. Numerical Results and Comparison …………………………………………….... 40

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS …………………………... 52

BIBLIOGRAPHY ………………………………………………………………………... 53

LIST OF TABLES

Table 4.1: Comparison of objective function value for data samples of size 10 42

Table 4.2: Comparison of objective function value for data samples of size 50 ……………… 42

Table 4.3: Comparison of objective function value for data samples of size 100 …………….. 43

Table 4.4: Comparison of objective function value for data samples of size 1000 …………… 44

Table 4.5: Comparison of objective function value for data samples of size 1500 …………… 44

Table 4.6: Comparison of objective function value for data samples of size 2000 …………… 45

Table 4.7: Comparison of computational time for data samples of size 10 …………………… 45

Table 4.8: Comparison of computational time for data samples of size 50 …………………… 46

Table 4.9: Comparison of computational time for data samples of size 100 ………………….. 46

Table 4.10: OA results in 60 seconds for data samples of size 100 …………………………… 47

Table 4.11: Objective value based on matrix $\Sigma = I$ …………………………………………..... 48

Table 4.12: Objective Values based on small k ……………………………………………….. 50

Table 4.13: Computational time based on small k …………………………………………….. 50

Table 4.14: Objective Values based on small k and big R …………………………………….. 51

LIST OF FIGURES

Figure 1.1: Systematic and Unsystematic risk ……………………………………………….... 5

Figure 1.2: Portfolio efficient frontier curve …………………………………………………... 6

Figure 1.3: Capital Market Line ……………………………………………………………….. 7

Figure 1.4: Characteristic Line 9

Figure 2.1: The Outer Approximation algorithm ……………………………………………... 29

Figure 4.1: Objective value trend …………………………………………………………….. 48

LIST OF SYMBOLS AND ABBREVIATIONS

BB	Branch-and-Bound
CAPM	Capital Asset Pricing Model
CML	Capital Market Line
CVaR	Conditional Value at Risk
GMD	Gini Mean Difference
HB	Heuristic Branching
MAD	Mean Absolute Deviation
MINLP	Mixed Integer Nonlinear Programming
MIQP	Mixed Integer Quadratic Programming
MV	Mean-Variance
NLP	Nonlinear Programming
OA	Outer Approximation
QP	Quadratic Programming
VaR	Value at Risk

CHAPTER 1 INTRODUCTION

1.1. Context

One of the most important concerns of investors of all time is to choose the best investing opportunities to maximize the value of their investment. Making decision on investment options is very important, challenging and complex especially for large financial institutions such as banks, insurance, investing and commercial institutions, real states and public sectors who seeks for high return opportunities at a reasonable risk. There is a variety of investing options that one might consider like stocks, bonds, gold, etc. but none of them is the best choice ever that means a rational person looks for a combination to spread the risk of loss which is known as *diversification*. A diversified portfolio has a smoother risk behavior that is less variation in expected return. Risk arise from uncertainty of data like future investment return which is forecasted. Well-diversification helps to decrease volatility of portfolio performance since, assuming that a portfolio assets are normally distributed, then the price of all assets does not change in the same direction at the same time and at the same rate otherwise the portfolio won't be well-diversified (Mansini et al., 2015; Moyer et al., 2006).

There are two sides for an investment namely *Risk* and *Return*. As a general rule in economy, one who seeks more return must expect more risk too and vice versa. An investor can be classified in one of the three categories; risk-averse is someone who avoid from taking risk thus so conservative, risk-taker on the contrary is ready to take more risk hoping to gain more return and risk-indifferent who is neither risk-averse nor risk-taker. As a matter of fact, although risk-takers are receptive for more risk but they all, as a rational person, requires a certain level as the minimum accepted return on investment for safety (Brigham & Houston, 2007).

The objective of financial decision makers is to maximize the value of investment projects for its owners that is to maximize shareholders' wealth which is measured by the market price of investing options say common stock. Market value (price) is defined as the price at which the stock trades in the marketplace. In other words, the higher return does not mean more value on investment due to associated risk. Risk and return are part and parcel of investment that has to be considered simultaneously (Brigham & Ehrhardt, 2008).

1.2. Organization of the thesis

This thesis composed of the following chapters to cover the relevant literature and proposed methodological approach to deal with the topic of this thesis. Here is a brief of what has been discussed in each chapter.

Chapter (1) begins with the introduction to the context of the problem and its affiliated attributes following by the contribution of the thesis. The rest of the chapter is devoted to a concise definition of the financial market, its types and effects to the problem at hand. At the end, the concept of risk and return and their correlation will be explained as well as the concept of diversification.

Chapter (2) will collect and compile the relevant literature in field of portfolio management containing portfolio optimization problem, portfolio optimization with real features in the market which mainly focuses on the two complicating features so called Cardinality constraint and bounds on investment that are the case of our problem. In the following, the most important methods to deal with the problem will be introduced addressing their pros and cons. A class of optimization problems named MINLPs and different approaches to solve them are discussed comprising some applicable and efficient bounding method for MINLP to generate a lower and upper bound to limit the domain of optimal value. Finally, some techniques such as adding valid inequalities and branching rules for tightening the gap between lower and upper bound thus improving the solution precision are investigated.

Chapter (3) will discuss about required methods and algorithmic approaches to deal with our problem. These methods include Outer Approximation algorithm and proposed Heuristic Branching algorithm compared with Branch-and-Bound as an exact global solution using CPLEX.

Chapter (4) is devoted to compare the numerical results of the proposed heuristic branching algorithm and outer approximation method versus most applicable methods from literature in terms of optimal value and time budget. All the results are obtained by executing random data that empirically tested.

1.3. Contribution

This research is focused on the Markowitz Mean-Variance (MV) portfolio optimization problem with cardinality constraints and bounding on variables so called as modern portfolio optimization problem which is a MINLP problem and well known as an NP-Hard problem. The problematic is that exact methods like branch and bound/cut solved even by CPLEX are not able to solve large instances in a reasonable time due to complexity of covariance matrix structure. In other words, as more securities included in portfolio, it increases the calculations geometrically since for the covariance matrix the number of correlation coefficient to be considered is $n(n-1)/2$ independent entries thus a large number of combination has to be computed to choose the best uncorrelated assets from the covariance matrix. Such complexities and the need for choosing optimal portfolio in a reasonable time in real stock market in which the transactions have to be fast, requires an efficient method to solve the portfolio selection problem considering a trade-off between solution quality and computational time which is the aim of this research.

The contribution of this research is twofold: First, a heuristic branching algorithm is proposed such that by finding a lower and upper bound and applying branching rules based on a cut, it produces near-optimal solution with acceptable precision and in a very timely manner which is suitable for large instances. Second, an Outer Approximation algorithm has been applied for the first time which shows very competitive results with exact methods. Finally, the results are compared with the other methods in literature and the best methods in different scales are classified. All the results have been empirically tested.

1.4. Financial Market

A financial market is a market in which people trade financial securities, commodities, and other fungible items of value at low transaction costs and at prices that reflect supply and demand. Securities include stocks and bonds, and commodities include precious metals or agricultural products. In economics, typically, the term market means the aggregate of possible buyers and sellers of a certain goods or service and the transactions between them. A securities market is used in an economy to attract new capital, transfer real assets in financial assets, determine price which will balance demand and supply and provide a means to invest money both short and long term (Brigham & Houston, 2007; Brigham & Ehrhardt, 2008).

1.5. Risk and Return

The **risk** is defined as the possibility that actual future returns will deviate from expected returns. In other words, it represents the variability of returns. From the perspective of security and investment analysis, risk is the *possibility that actual cash flows (returns) will be different from forecasted cash flows (returns).* An investment is said to be *risk-free* if the money returns from the initial investment are known with certainty. Some of the best examples of risk-free investments are U.S. Treasury securities. There are two categories of risks; systematic and unsystematic. Systematic (Market) risk arise from overall economic and industry condition and is inherent in the nature of the business thus it is beyond management control and con not be reduced. Unsystematic risk, from the other side, is a result of miss-management, low forecasting accuracy or any other shortcoming in the process of planning or decision making which can be reduced by more logical and correct decisions. Consequently, unsystematic risk is under control, although it con not be totally eliminated but it could be substantially decreased (see figure 1.1) (Moyer et al., 2006).

1.6. Portfolio diversification

Diversification is a way to lessen unsystematic risk in portfolio. The idea behind diversification is to spread investment budget over a set of assets to create a portfolio of diverse assets hence spreading the risk. The logic of holding diverse assets is that the price of diverse assets does not change in the same direction and at the same time or at the same rate. Technically, combining assets with negatively correlated returns and lower correlations, significantly reduces the total variability or dispersion (i.e. risk) of portfolio return. In other words, “do not put all your eggs in one basket”.

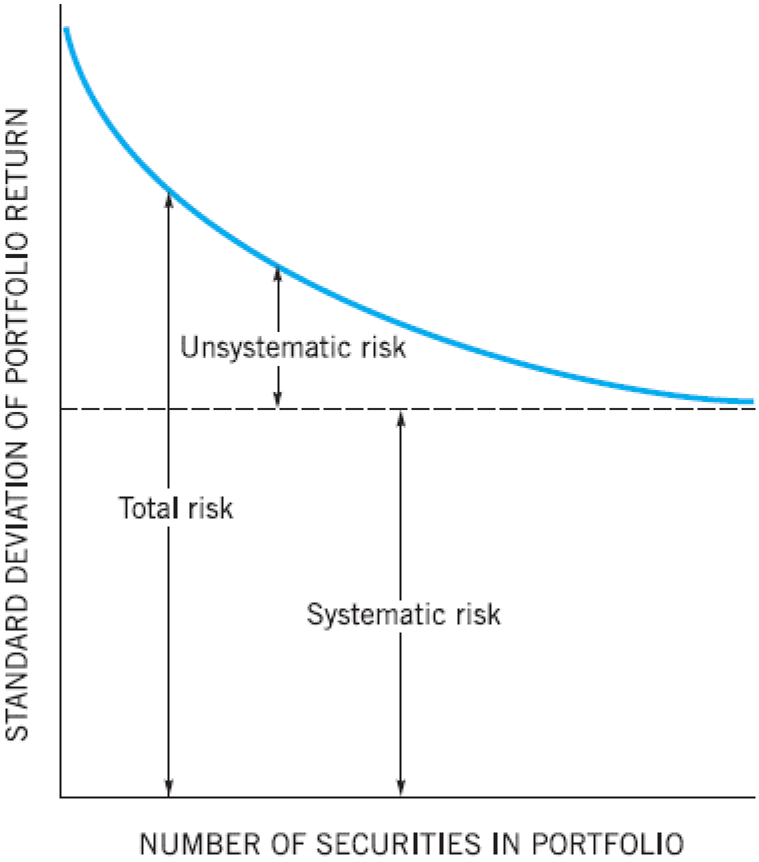

Figure 1.1: Systematic and Unsystematic risk

A portfolio called efficient if for a given expected return, no other portfolio has a lower risk and vice versa that is for a certain level of risk, no other portfolio has higher expected return. The curve which connect the efficient portfolios A and B as shown in figure 1.2 is called efficient frontier. Selecting optimal portfolio whether to maximize return or minimize risk, depends on the investor’s degree of risk reception. More conservative (risk averse) investors choose their optimal portfolios among efficient portfolios (Moyer et al., 2006).

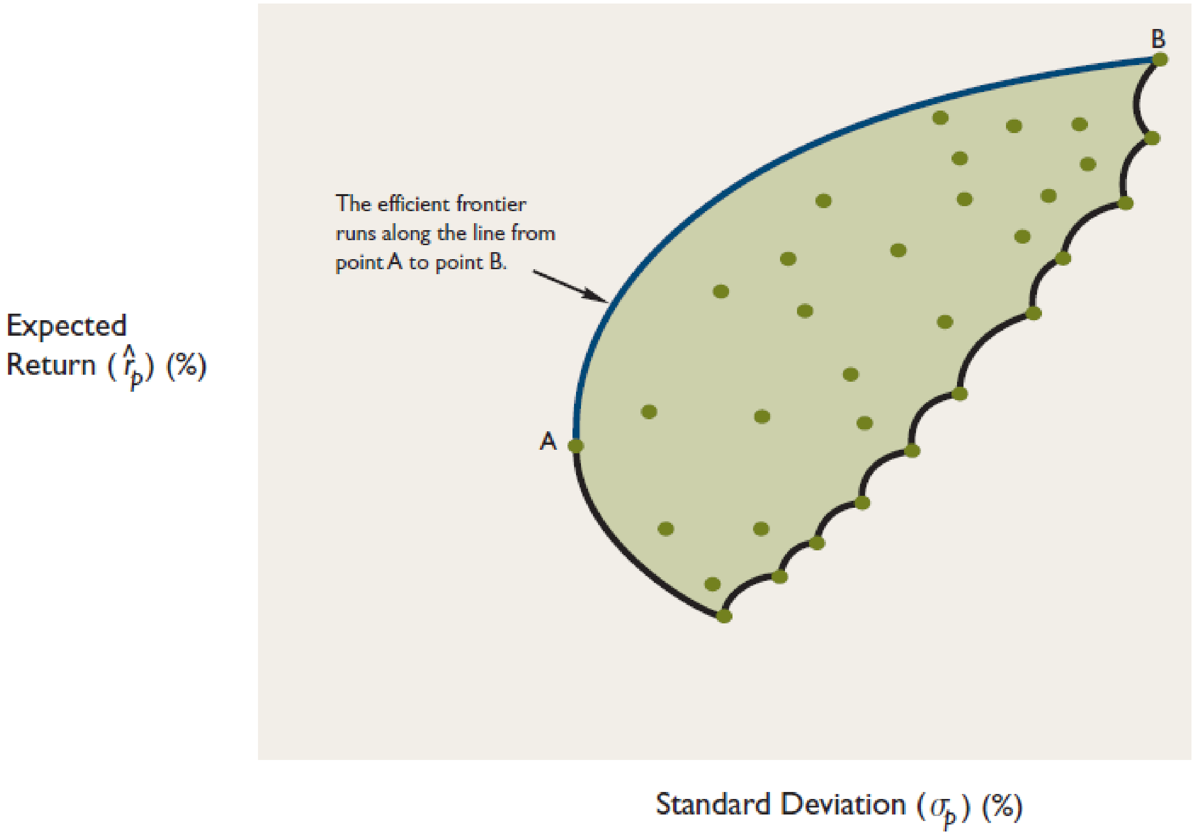

Figure 1.2: Portfolio efficient frontier curve (Moyer et al., 2006)

Evaluating portfolio risk performance when there are more securities included increases the calculations geometrically since for the covariance matrix the number of correlation coefficient to be considered is n (n - 1) /2 independent entries thus a large number of sample has to be computed to choose the best uncorrelated assets from the covariance matrix. (Tapiero, 2004; Mansini et al., 2015).

Let r_f denote the risk-free rate of return and m the market portfolio with market rate of return r_m and standard deviation σ_m as market risk in state of market equilibrium. The line connecting r_f and m as shown in figure 1.3 is called *capital market line* (CML). The slope of the capital market line measures the portfolio market price at equilibrium. If investors are able to lend and borrow money at the rate r_f, then any risk-return combination on the capital market line can be achived by investing (i.e. lending) a pert of money on risk-free assets and the rest in portfolio m (between r_f and m on the CML) or by borrowing money at the rate r_f and investing them in portfolio m (above m on the CML). Risk-averse investors tend to minimize risk and invest at around point r_f and more risk-takers around point m on the capital market line hoping to gain more return (Moyer et al., 2006).

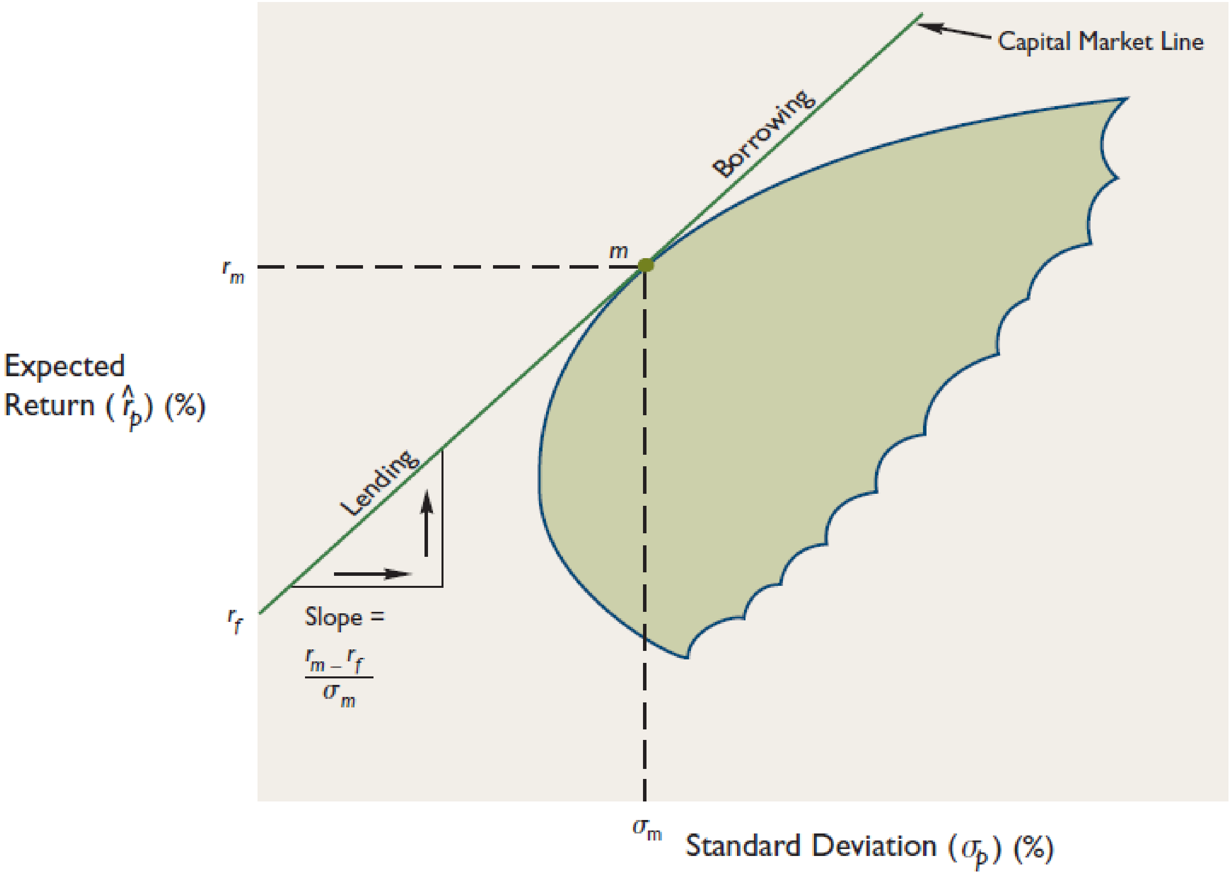
Capital Market Line
Borrowing
m
r_m
Expected Return ($\hat{r}_p$) (%)
Lending
r_f
Slope = $\frac{r_m - r_f}{\sigma_m}$
σ_m
Standard Deviation (σ_p) (%)

Figure 1.3: Capital Market Line (Moyer et al., 2006)

A widespread method to analyze the relationship between risk and return is Capital Asset Pricing Model (CAPM). However, all models are subject to data estimation errors, and possibly to modeling errors.

1.7. Capital Asset Pricing Model (CAPM)

$$Cost\ of\ Equity = R_f + \beta(R_m - R_f)$$

R_f is the rate of return on a risk-free investment (i.e. Government bonds or Treasury Bills). Treasury bill is Short-term security sold by a central bank to meet a government’s short-term financial requirements. Treasury bills are easily sold and have a relatively low rate, but they are nearly risk-free and are exempt from taxes.

β is a measurement of the relative risk of a specific share to the market as a whole. (A beta share greater than 1 is considered to have greater risk than the market, and a beta share less than 1 is considered to have less risk than the market.)

R_m is the average rate of return on the market as a whole over the period being analyses.

(R_m - R_f) represents the market risk premium, or the difference in rates of return between investments that are considered risk-free, such as government bonds, and those investments that are market driven, over the period being analyses.

Risk premium [$\beta(R_m - R_f)$]: Extra return expected from a risky asset compared with the return from an alternative risk-free asset.

Notice: β is Systematic (Market) Risk that cannot be eliminated by portfolio diversification.

(Brigham & Houston, 2007; Brigham & Ehrhardt, 2008).

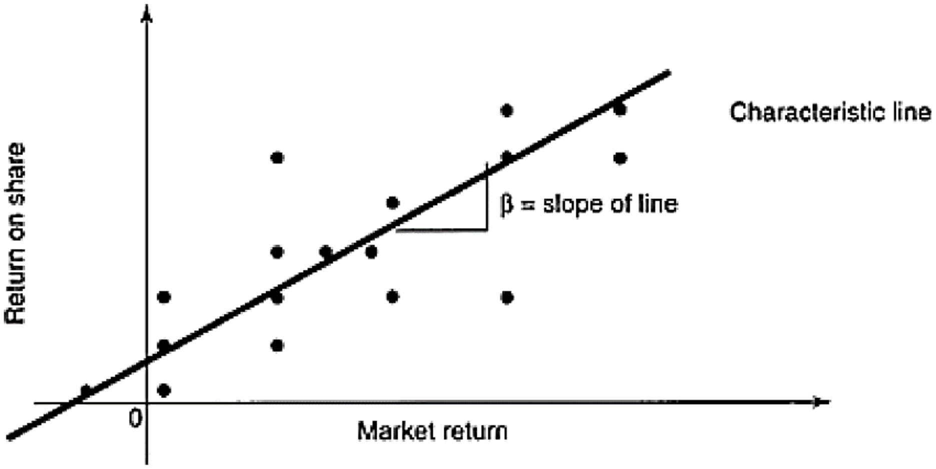

Figure 1.4: Characteristic Line

β = the slope of Regression line or $\beta = \frac{Cov_{sm}}{Var_m}$

Portfolio beta: $\beta_p = \sum_{i=1}^{n} x_i\, \beta_i$, $(\sum x_i = 1)$

Portfolio return: $R_p = \sum_{i=1}^{n} x_i r_i$, $(\sum x_i = 1)$

(x_i : proportion of share *i* in a portfolio)

Therefore, if a particular security's price movement is greater than the prescribed benchmark, a value of β > 1 is given. This constitutes a volatile or risky security, whereas β < 1 is treated as an investment that involves a low amount of risk (Stoyan, 2009).

CHAPTER 2 LITERATURE REVIEW

2.1. Portfolio Optimization Problem

An investor seeks to invest his money in a number of different securities (stock, bonds, etc.) such that having minimum risk for a certain return. There are two main types of risks associated with investment; (1) the risk of having less return as expected for securities, (2) the risk of not well-diversified portfolio, which stems from not having enough different types of securities in portfolio. Consider an investor who has a certain amount of money to be invested in a number of different securities (stocks, bonds, etc.) with random returns. Let μ_i be expected return and σ_i^2 covariance for each security $i = 1, \cdots, n$. In addition, for any two securities *i* and *j*, their correlation coefficient ρ_{ij} is also assumed to be known. Let x_i represent the proportion of the total funds invested in security *i*, then the expected return and the variance of the portfolio $x = (x_1, \cdots, x_n)$ are as follows

$$E[x] = x_1\mu_1 + \cdots + x_n\mu_n = \mu^T x$$

$$Var[x] = \sum_{i,j} \rho_{ij}\, \sigma_i\sigma_j x_i x_j = x^T \Sigma\, x$$

Where $\rho_{ii} \equiv 1$, $\Sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ and $\mu = (\mu_1, \cdots, \mu_n)$.

A feasible portfolio x is called efficient if with a certain amount of risk, it provides the maximum expected return or, vice versa, having a certain expected return among all securities, it has minimum risk value. The collection of efficient portfolios forms the efficient frontier of the portfolio universe (Cornuejols & Tutuncu, 2006; Prigent, 2007).

The standard portfolio optimization problem model known as the Markowitz' Mean-Variance portfolio optimization model can be formulated in three equivalent ways as bellow;

(1) The first model aims to minimize the portfolio variance for securities 1 to *n* respecting at least a target value *R* as expected return. In this case, the objective is a convex quadratic function over a set of linear constraints:

$$\begin{aligned}\min_{x}\ & x^T \Sigma\, x \\ & \mu^T x \geq R \\ & e^T x = 1 \\ & x \geq 0\end{aligned}$$

where e is an n-dimensional sum vector (all of its components are equal to 1). The first constraint requires a target value as the minimum expected return. The second constraint indicates that the investment proportions in different securities x_i should sum to 1. Non-negativity constraints on x_i are introduced to prevent short sales (selling a security that you do not have).

(2) The second model tries to maximize the expected return of portfolio considering a tolerance limit for portfolio risk.

$$\begin{aligned}\max_{x}\ & \mu^T x \\ & x^T \Sigma\, x \leq \sigma^2 \\ & \textstyle\sum_{i=1}^{n} x_i = 1 \\ & x \geq 0\end{aligned}$$

(3) The last model combines the expected return and weighted risk as objective function.

$$\max_{x}\ \mu^T x - \lambda\, x^T \Sigma\, x$$

$$x \in C = \{x \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = 1, x \geq 0\}$$

Where λ is the investors risk aversion coefficients vector. Since variance is always non-negative, then

$$x^T \Sigma\, x \geq 0 \qquad \forall\, x$$

which means Σ is positive semi-definite.

The all three models proposed above have a worst-case orientation. That is, we try to optimize the behavior of the solutions under the most adverse conditions (Cornuejols & Tutuncu, 2006).

2.2. Portfolio Optimization with Real Feature

In practice, investors have different preferences or limitations to be considered in portfolio selection which stem from real features in the market that might not be taken into account in the model such as restricting the number of assets selected (*cardinality constraint*), bounding on the amount of money invested in an asset (*bound constraints*), transaction costs and decision dependency requirements (*logical constraints*) (Mansini et al., 2015; Maringer, 2005).

2.2.1. Cardinality Constraints

According to modern portfolio theory, the investors aimed to minimize risk with a certain level of return. A well-diversified portfolio reduces unsystematic risk which is influenced by decision so it is under control through logical decision making. Investing in many assets with small amount will increase the total risk as well as total cost due to fixed transaction costs associated to the number of assets in portfolio incurred when buying or selling assets. On the other side, investing on a few assets increases the opportunity loss cost of not choosing the best combination out of all assets also not well spreading risk over assets thus not a well-diversified portfolio. As a result, investors prefer to restrict the number of assets to be held in portfolio. A restriction on the number of assets that can be selected in a portfolio is called *cardinality constraint* (Mansini et al., 2015).

In order to impose the cardinality constraint in a portfolio optimization model first we define binary variables $y_i : \; i = 1, \dots, n$, where y_i is equal to 1 if asset i is selected in the portfolio, and 0 otherwise. The cardinality constraint can be expressed by restricting the number of assets in portfolio not to be greater than a predefined number K_{max} and not lower than a minimum number K_{min}:

$$K_{min} \leq \sum_{i=1}^{n} y_i \leq K_{max}$$

The cardinality constraint is usually accompanied by the bound constraints on investment.

2.2.2. Bounds on Investment

A very practical limitation in financial decision making is to enforce decision variables to take values within a given interval by applying lower and upper bound limit on the share x_i or on the amount of an asset held in the portfolio. These constraints are also known as *threshold constraints.* Such a constraint is formulated as:

$$l_i \leq x_i \leq u_i \qquad \forall\, i = 1, \ldots, n$$

If the bounds on investment is applicable only if the asset is selected in the portfolio, the above constraint will be modified using the binary variables y_i as follow (Mansini et al., 2015).

$$l_i y_i \leq x_i \leq u_i y_i \qquad \forall\, i = 1, \ldots, n$$

2.3. Problematic

The classical Mean-Variance (MV) portfolio selection model is the pillars of modern portfolio theory. However, there has been several criticism to model addressing the nonrealistic assumptions such as normal behavior of portfolio risk and non-well-diversified securities in portfolio.

One of the most important refinements that have been proposed to make the model more realistic is to limit the number of assets to be held in an efficient portfolio so-called cardinality constraint. The other refinement recommends lower and upper bounds on the proportion of the investment in each asset named as quantity constraints. The last requirement comes from the truth that a very small investment in some securities is not beneficial because of transactions costs. The Markowitz model with the above requirements is called Limited Asset Markowitz Model (LAM).

According to Cesarone et al (2010), in order to define the LAM model, the realistic constraint that at maximum K assets should be held in the portfolio (called cardinality constraint) and the quantity of x_i for each asset that is included in the portfolio should be limited within a given interval $[l_i\ , u_i]$ (a quantity constraint or buy-in threshold), will be added to Markowitz classical model as Model 2.1:

Model 2.1

$$\min_{x} \; x^T \Sigma \, x$$

S.t.

$$\mu^T x \geq R$$

$$\sum_{i=1}^{n} x_i = 1$$

$$x_i = 0 \;\; or \;\; l_i \leq x_i \leq u_i \, , \quad i = 1, \dots , n$$

$$|supp(x)| \leq k$$

Where $supp(x) = \{i: x_i > 0\}$.

The Model 2.1 has a convex quadratic objective function but the set of constraints is not convex anymore. This problem can be reformulated as a Mixed Integer Quadratic Program (MIQP) with the addition of *n* binary variables as Model 2.2:

Model 2.2

$$\min_{x} \; x^T \Sigma \, x$$

S.t.

$$\sum_{i=1}^{n} \mu_i \, x_i \geq R$$

$$\sum_{i=1}^{n} x_i = 1$$

$$\sum_{i=1}^{n} y_i \leq k$$

$$l_i y_i \leq x_i \leq u_i y_i \, , \quad i = 1, \dots , n$$

$$x_i \geq 0 \qquad i = 1, \dots , n$$

$$y_i \in \{0,1\} \qquad i = 1, \dots , n$$

Such an MIQP model is classified as an NP-Hard problem (Cesarone et al, 2010).

According to Cesarone et al (2010), “A number of exact approaches have been proposed to solve the problem above. Bienstock (1996), proposes a branch-and-cut algorithm and reports good computational results for some real-life problems; however, his method seems to become extremely slow for small values of K. Since exact methods are able to solve only a fraction of practically useful LAM model, a variety of heuristic procedures have also been proposed for solving the problem."

Different approaches have been proposed to simplify portfolio selection model like approximating the quadratic objective function with linear one. These approaches make use of the approximation or the decomposition of the covariance matrix. Some other studies applied a different linear function, which leads to the same goal. For example, using the mean absolute deviation (MAD) model. One another important risk measurement tool is Value-at-Risk (*VaR*), but the VaR optimization problem is not convex, also it does not benefit from diversification. However, Conditional Value at Risk (*CVaR*) has been introduced which resolves difficulties of VaR but still it is an approximation to the original function.

The following will shortly review the LP optimization model which outperforms among the others and will be compared with the proposed heuristic method in this research.

- Mean Absolute Deviation (MAD)
- Gini Mean Difference (GMD)
- Conditional Value at Risk (CVaR)

2.4. Mean Absolute Deviation (MAD)

An alternative to the standard deviation (SD) as common error measurement is the Mean Absolute Deviation (MAD), which provide a linear function however, its value is always less than SD since it does not consider the correlation among variables.

$$MAD = E\left|\sum_{j=1}^{n} \mu_j x_j - E\left(\sum_{j=1}^{n} \mu_j x_j\right)\right|$$

For normally distributed variables the following ratio holds

$$\frac{MAD}{SD} = \frac{E|X|}{\sqrt{E(X^2)}} = \sqrt{\frac{2}{\pi}} = 0.79788$$

It has been shown that if the returns are multivariate normally distributed, the minimization of the MAD provides similar results as the classical Markowitz formulation. However, they did not consider cardinality as a discrete model formulation. Also, the assumption "*multivariate normally distributed"* returns does not always hold since, in order to reduce unsystematic risk, a portfolio has to be well diversified which means it should include some complementary and diverse securities. Thus, there exist interrelation among securities and some securities might dominate some others. Such conditions and more does not guarantee the "*multivariate normally distributed"* assumption (Konno &Yamazaki, 1991) see also (Rudolf et al., 1999).

2.5. Gini Mean Difference (GMD)

Another dispersion metric as a risk measurement for random variables which are independently and identically distributed with the same (unknown) distribution is the Gini's Mean Difference between each pair of variables.

$$GMD = E\left|x_i - x_j\right| = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right| \qquad : \; i \neq j$$

Similar to MAD, if the rates of return are multivariate normally distributed, minimizing GMD is equivalent to minimizing standard deviation (Mansini, 2015).

2.6. Value at Risk (VaR)

One well-known risk measure is Value at Risk (VaR) which represents the predicted maximum loss with a specified probability level (e.g., 95%) over a given period of time. Given a confidence level $\alpha \in (0,1)$, the VaR of the portfolio at the confidence level α is given by the smallest number γ such that the probability that the loss X exceeds γ is at most $(1-\alpha)$. Mathematically, if X is the loss of a portfolio over a fixed period of time, then $VaR_\alpha(X)$ is the level α-quantile, i.e.

$$VaR_\alpha(X) = inf\{\gamma \in \mathbb{R} : P(X \geq \gamma) \leq 1-\alpha\} = inf\{\gamma \in \mathbb{R} : \Psi(x,\gamma) > \alpha\}$$

Where $\Psi(x, \gamma)$, is the cumulative distribution function. VaR has one important drawback that is, it lacks subadditivity property. In other words, risk measures should consider risk diversification and therefore, satisfy the following subadditivity property:

$$f(x_1 + x_2) \leq f(x_1) + f(x_2)$$

But, VaR of two different investment portfolios might be greater than the sum of the individual VaRs. Also, VaR is nonconvex and non-smooth and has multiple local minimum, which makes the problem difficult seeking the global minimum. Another criticism of *VaR* is that it pays no attention to the magnitude of losses beyond the *VaR* value. Due to above mentioned shortfalls of VaR a modified alternative named Conditional Value at Risk (***CVaR***) has been introduced.

CVaR also called "expected shortfall" at α% level is the expected loss given that the loss exceeds VaR and is an alternative to *VaR* that is more sensitive to the shape of the loss distribution in the tail of the distribution.

We denote the portfolio choice vector by x and the random events by the vector v that has a probability density function denoted by p(v). Let $f(x, v)$ denote the loss function. Then the $\alpha - CVaR$ for portfolio x is defined as:

$$CVaR_{\alpha}(x) = \frac{1}{1-\alpha} \int_{f(x,v) \geq VaR_{\alpha}(x)} f(x, v) p(v) d(v)$$

N.B. *CVaR* of a portfolio is always at least as big as its *VaR* that is $CVaR_{\alpha}(x) \geq VaR_{\alpha}(x)$.

For discrete probability distribution (where event v_j occurs with probability P_j for $j = 1, \cdots, n$) the *CVaR* becomes (Cornuejols & Tutuncu, 2006).

$$CVaR_{\alpha}(x) = \frac{1}{1-\alpha} \sum_{j: f(x, v_j) \geq VaR_{\alpha}(x)} p_j f(x, v_j)$$

2.6.1. Minimizing CVaR

In order to simplify the function above for optimization, the following function will be considered instead

$$F_\alpha(x,\gamma) := \gamma + \frac{1}{1-\alpha}\int_{f(x,v)\geq\gamma}(f(x,v)-\gamma)p(v)d(v)$$

Or equivalently

$$F_\alpha(x,\gamma) := \gamma + \frac{1}{1-\alpha}\int(f(x,v)-\gamma)^{+}p(v)d(v)$$

The following properties hold for the function above

1. The $F_\alpha(x,\gamma)$ is a convex function of γ
2. $VaR_\alpha(x)$ is a minimizer over γ of $F_\alpha(x,\gamma)$
3. The minimum value over γ of the function $F_\alpha(x,\gamma)$ is $CVaR_\alpha(x)$

As a result of these properties, it will be concluded that

$$\min_{x} CVaR_\alpha(x) = \min_{x,\gamma} F_\alpha(x,\gamma)$$

Since it is not desirable to compute density function *P(V)* of random events, we assume a number of scenarios $(v_i: i = 1,\cdots,S)$ with same probability. In this case, we obtain the following approximation to the function $F_\alpha(x,\gamma)$ by using the empirical distribution of the random events based on the available scenarios:

$$\hat{F}_\alpha(x,\gamma) \coloneqq \gamma + \frac{1}{(1-\alpha)S}\sum_{i=1}^{S}(f(x,v_i)-\gamma)^{+}$$

To solve the *CVaR* portfolio optimization problem using this function, artificial variables z_i are introduced such that

$$\begin{aligned} &\underset{x,z,\gamma}{\text{Min}}\ \ \gamma + \tfrac{1}{(1-\alpha)S}\textstyle\sum_{i=1}^{S} z_i \\ &\quad \text{S.t.} \\ &\quad z_i \geq (f(x,v_i)-\gamma) \qquad \forall\ i = 1,\cdots,S \\ &\quad z_i \geq 0 \qquad\qquad\qquad\ \ \forall\ i = 1,\cdots,S \\ &\quad x \in X \end{aligned}$$

(Cornuejols & Tutuncu, 2006).

2.7. Mixed Integer Nonlinear Programming Models

An optimization problem which has nonlinear functions and contains continuous and discrete variables together is known as Mixed Integer Nonlinear Program (*MINLP*). A general MINLP model could be defined as Model (2.3);

$$\begin{aligned} & Min \;\; f(x,y) \\ & s.t. \;\; g(x,y) \leq 0 \\ & \qquad h(x,y) = 0 \\ & \qquad x \in [\,\underline{x}\,, \overline{x}\,] \\ & \qquad y \in [\,\underline{y}\,, \overline{y}\,] \;\; integer \end{aligned} \qquad \text{Model (2.3)}$$

Where the vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{Z}^m$ are finite and *f*, g and *h*: $\mathbb{R}^n \rightarrow \mathbb{R}$ are nonlinear functions. In case that *f* and g are convex and *h* is affine, the MINLP is convex otherwise, it is nonconvex. This types of problems have vast applications in different fields like economy, finance, engineering, etc. (Nowak, 2005).

A restricted version of MINLP in which only the objective function is nonlinear and the constraints are all linear could be expressed by Model (2.4);

$$\begin{aligned} & Min \;\; f(x,y) \\ & s.t. \;\; G(x,y) \leq b \\ & \qquad H(x,y) = d \\ & \qquad x \in [\,\underline{x}\,, \overline{x}\,] \\ & \qquad y \in [\,\underline{y}\,, \overline{y}\,] \;\; integer \end{aligned} \qquad \text{Model (2.4)}$$

Where *G* and *H* are rational matrix and *b* and *d* are rational vectors. Although, Model (2.4) have simpler linear constraints but still is an MINLP. Both Models (2.3) and (2.4) are classified as NP-Hard problem due to combining nonlinearity and integrality in the model (Junger et al., 2010; Lee & Leyffer, 2012).

2.8. Fundamental methods to solve MINLP

Generally, MINLPs are challenging combinatorial optimization problems which combines integrality and nonlinearity in the problem that each of which arise complexity for the problem. If one removes the integrality from the problem but preserves the nonlinear functions, then MINLP reduces to General Nonlinear program (NLP) which could be NP-Hard. From the other side, restricting the model to contain only linear functions but maintaining integer variables, the MINLP reduces to Mixed-Integer Linear program (MILP) that are easier to solve but could be difficult especially for very large instances. However, under certain conditions NLP will be easier to solve efficiently. If in MINLP, the objective function and all constraints are convex and bounded, thanks to properties of convexity, relaxing integrality leads to a convex NLP problem for which there exist efficient algorithms (Lee & Leyffer, 2012).

From algorithmic and solvers perspective, there is a giant gap between MINLP and MIP. With state-of-the-art solvers, an MIP could be solved for large scale problems even with millions of variables and constraints (not in the worst case) while the dimension of solvable MINLPs is often limited by a number that is smaller by three or four orders of magnitude. One may consider a linear approximation of nonlinear functions to transform MINLP to MILP (like piecewise linearization which would be explain further in section 2.8.1) but these approximations are usually poor with low precision and in some cases, it ends up to a very large MILP which could be NP-Hard problem (Nowak, 2005).

There are variety classes of methods to deal with MINLP. However, most of methods apply some form of tree-search. Here is a brief introduction to these class of methods. The ***first*** class is so-called Decomposition Methods used to solve large scale problems with separable structure. In this class, if the problem has a block-separable structure, it can be split into several smaller sub-problems each of which corresponds to a separate block of the original MINLP. Sub-problems often hold specific complicated constraints and/or variables and their solutions will be combined through joining (common) constraints in a *Master* problem which is usually a simple problem containing general constraints. Decomposition methods differ based on their underlying principles and problem formulation. Such methods are mainly: Dantzig-Wolfe Decomposition, Column Generation, Benders Decomposition, Dual methods – Lagrangian Decomposition and Primal cutting-plane methods (Nowak, 2005).

The ***second*** class is based on generating and tightening bounds on the optimal solution of the MINLP. In this class, algorithms solve a sequence of updated sub-problems to generate and improve bounds iteratively until certain stopping criterions has been reached. Here, two fundamental concepts are: *Relaxation* and *Constraint Enforcement*. Considering a Minimization problem, Relaxation is used to obtain a *lower bound* for the problem by dropping integrality from variables or eliminating some constraints resulting in larger feasible set. If all the remaining constraints after relaxation are convex then the feasible set is convex which makes it easier to solve. An *upper bound* on the optimal solution could be attain from any feasible or heuristic solution. Since, basically heuristic solutions are sub-optimal (does not guarantee global optimality) it can be considered as an upper bound for minimization problem. Now one can apply efficient decisions to tighten either lower or upper bound to approach each other until reaching a given tolerance as termination rule. Constraint enforcement is aimed at excluding the part of feasible solution from the feasible set after relaxation. The feasible solution to be excluded are feasible for the relaxed problem but not for the original MINLP. Constraint enforcement basically carried out by adding *Valid inequalities* like valid cuts to update and tightening bounds or by branching scheme which divides relaxation problem into two or more sub-problems (Belotti et al., 2012).

The ***third*** class is so-called meta-heuristic. Some complicated real world problems especially NP-Hard problems cannot be solved to global optimality with the state-of-the-art methods due to complexity or large sizes as well as expensive computational time. However, we are interested to obtain a good solution in a timely manner. In such cases, heuristic methods which does not guarantee optimality but accelerates computations are used. Heuristics differ in search techniques and most of them take advantage of probabilistic search schemes that require a random choice of a candidate solution at each iteration to escape from local optimality trap. Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO), Particle-Swarm Optimization (PSO), etc. are fall under this category (Belotti et al., 2012).

2.8.1. Piecewise Linearization

A very first idea to deal with nonlinear functions is to make use of a series of piecewise linear function to approximate a nonlinear function. To do so, an interval [a, b] for the nonlinear function *f(x)* will be subdivided into smaller intervals with a linear function for each adjacent pairs of points. Obviously, the more piecewise linear functions, the better and more precise approximation of a nonlinear function. In high dimension the same principle applies like in *n*-dimension, lines are simplices. Here the difficulty reveals that is the number of simplices needed increases drastically as dimension increases. Thus, using piecewise linearization seems applicable for MINLPs merely if the nonlinear functions have only a few variables and/or smoother function (Lee & Leyffer, 2012).

In this paper the focus is on the second-class methods to cope with MINLP. The following are some relevant and more applicable techniques to find a lower and upper bound for the MINLP and some valid cuts for tightening the bounds on the problem.

2.8.2. Lower Bounds

In this session, some techniques to generate a lower bound, in a relatively cheap way, for general MINLP are discussed. Such a lower bound is essential for branch and bound (BB) type algorithms and are mainly computed through relaxation or underestimations.

Definition 2.1. given an optimization problem

$$\min_{x \in S} f(x)$$

A relaxation for *f(x)* is any optimization problem of the form

$$\min_{x \in \check{S}} \check{f}(x)$$

Such that

$$S \subseteq \check{S};$$

$$x \in S \implies \check{f}(x) \leq f(x).$$

where $\check{f}(x)$ is a lower bound for *f(x)*. If $\check{S}$ is a convex set and $\check{f}$ is convex on $\check{S}$, then we have a convex relaxation which can be solved efficiently to optimality (Locatelli & Schoen, 2013).

Integer Variable Relaxation

One way to relax the MINLP is to drop the integrality that is variable $y \in \mathbb{Z}$ will be relaxed to $y \in \mathbb{R}$ which reduces MINLP to NLP. Since, relaxation enlarges the feasible set the optimal solution obtained is a lower bound for the original problem.

Constraint Relaxation

Eliminating the constraints containing integer variables from the original model will result in feasible set enlargement and consequently the solution obtained is a lower bound for the original problem but such a relaxation often provides a loose lower bound.

LP-bound

If the integrality is dropped and all nonlinear functions underestimated linearly using any linear approximation (say Taylor linear approximation) then MINLP reduces to LP model that its solution is a lower bound for the MINLP. However, the quality of LP-bound depends on the preciseness of the linear underestimation of nonlinear functions. This type of relaxation for MINLP especially for nonconvex functions usually produces weak bounds (Nowak, 2005).

α-underestimator

Having a continuously twice-differentiable quadratic function $f(x) = x^T Qx + C^T x$ over $x \in [\, \underline{x}\, , \overline{x}\,]$, a lower bound such that $\check{f}(x) \leq f(x)$ could be attained by

$$\check{f}(x) = f(x) - \alpha \sum_{i=1}^{n} (\, x_i - \underline{x}_i\,)(\, \overline{x}_i - x_i\,)$$

Where $\alpha \geq 0$. For a closer lower bound it is sufficient to set $\alpha = \lambda_{min}(Q)$ means the minimum eigenvalue of Q, to obtain a convex relaxation. This method can be extended to non-quadratic functions while preserving the validity of the lower bound (Belotti et al., 2012; Nowak, 2005).

2.8.3. Upper Bounds

A class of heuristic methods usually applied to obtain a feasible point as an upper bound for MINLP problem in a timely manner. Some of these heuristics may completely ignore the objective function and focus on finding only a feasible solution.

MILP-based rounding

Finding a locally optimal solution to the continuous relaxation of the MINLP is usually easier and computationally faster than solving the MINLP itself. Given a solution $\hat{x}$ of the continuous relaxation, one can try rounding fractional values of integer-constrained variables to obtain a feasible solution around $\hat{x}$ as an upper bound. However, such a rounding may end up with an infeasible solution (Belotti et al., 2012).

Feasibility pump

The main idea, like that in MILP based rounding described above, is that an NLP solver can be used to find a solution that satisfies nonlinear constraints. Integrality is enforced by solving an MILP. An alternating sequence of NLP and MILP is solved that may lead to a solution feasible to the MINLP. The main difference from the rounding approach is the way MILP is set up (Belotti et al., 2012).

Undercover

The Undercover heuristic is specially designed for nonconvex MINLPs. The basic idea is to fix certain variables in the problem to specific values so that the resulting restriction becomes easier to solve (Belotti et al., 2012).

2.8.4. Valid cuts

Valid cuts are like redundant constraints that do not cut off any part of feasible region of the original problem which contain potential solution points. In fact, valid cuts used to cut out a portion of relaxed problem that does not hold solution points in order to tighten the lower / upper bound in a repetitive manner thus improving the solution. Valid cuts are divided into general and specific cuts. General cuts can be used for most of the problems but basically, they are not as efficient as specific cuts which are defined specifically for a given problem. A common idea is to use valid inequalities to derive branching rules or selecting the most violated cut at the optimal solution of the relaxation problem to be added to the model to tighten the optimality gap or making decision on variables.

Definition 2.2.

An inequality $\pi^T x \leq \pi_0$ is a valid inequality for $X \subseteq \mathbb{R}^n$ if $\pi^T x \leq \pi_0$ is satisfied by all $x \in X$.

Mixed-integer Rounding Cuts

This types of cut are added iteratively to remove fractional solutions from relaxation problem. Let’s define the set *S:*

$$s := \{(x, y) \in \mathbb{R}\times\mathbb{Z} \mid y \leq b + x\ , x \geq 0\}$$

Let $f_0 = b - \lfloor b \rfloor$, then the inequality

$$y \leq \lfloor b \rfloor + \frac{x}{1 - f_0}$$

is valid for *x* by verifying it for two cases: $y \leq \lfloor b \rfloor$ and $y \geq \lfloor b \rfloor + 1$ (Belotti et al., 2012). For more detail see (Wolsey, 1998).

Perspective Cuts for MINLP

In many problems, a binary variable is utilized to model a continuous variable upper bound. If *y* and *x* are binary and continuous variable respectively, the relationship is as follow:

$$x \leq uy$$

According to Belotti et al., (2012) if the continuous variable shows up in a convex nonlinear constraint, then a reformulation technique known as *perspective* cut can be used for strengthening relaxation. One can define the set *S:*

$$S = \{(x_1, x_2, y) \in \mathbb{R}^2 \times \{0,1\} : \ x_2 \geq x_1^2, \ uy \geq x_1 \geq 0\}$$

Such that the set *S* is the union of two convex sets: $S = S^0 \cup S^1$, where

$$S^0 = \{(0, x_2, 0) \in \mathbb{R}^3 : \ x_2 \geq 0\},$$

$$S^1 = \{(x_1, x_2, 1) \in \mathbb{R}^3 : \ x_2 \geq x_1^2, \ u \geq x_1 \geq 0\}.$$

Now the convex hull of *S* can be defined as

$$conv(S) = \{(x_1, x_2, y) \in \mathbb{R}^3 : \ x_2 y \geq x_1^2 , \ 0 \leq x_1 \leq uy, \ 0 \leq y \leq 1, \ x_2 \geq 0\}$$

“The term $x_2 y \geq x_1^2$ plays the role of *perspective function*. For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, the perspective function $P : \mathbb{R}^{n+1} \to \mathbb{R}$ of f is

$$P(x,z) \coloneqq \begin{cases} 0 & if\ z = 0, \\ zf(x/z) & if\ z > 0. \end{cases}$$

If *z* indicates binary variable that push variable $x = 0$ otherwise, the convex nonlinear constraint $f(x) \leq 0$ must hold, then replacing the constraint $f(x) \leq 0$ with $zf(x/z) \leq 0$, results in a convex inequality that significantly tightens the relaxation of feasible region.”

(Notice: This section has been directly quoted from Belotti et al., 2012)

Disjunctive Inequality

Consider two sets of constraints that we are interested to satisfy one of them, then an inequality called *disjunctive inequality* is used for the union of two sets.

If $\sum_{j\in N}\pi_j^1 x_j \leq \pi_0^1$ is valid for $S_1 \subset R_+^n$ and $\sum_{j\in N}\pi_j^2 x_j \leq \pi_0^2$ is valid for $S_2 \subset R_+^n$, then

$$\sum_{j\in N} min\left(\pi_j^1, \pi_j^2\right) x_j \leq max(\pi_0^1, \pi_0^2)$$

is valid for $S_1 \cup S_2$ (Nemhauser & Wolsey, 1999).

Level Cuts

Level cuts are based on the idea that in each iteration the objective function value shall not exceed its previous iteration value. For $min\{f(x) \mid g(x) \leq 0\}$, if $\bar{z}$ indicate an upper bound of the optimal value which can be obtained by $f(\hat{x})$ at a feasible point $\hat{x}$ or by maximizing *f(x)* over convex relaxation of the feasible set, then the *linear level cut* bellow

$$f(x) \leq \bar{z}$$

is a valid cut for the problem. A *nonlinear level cut* also can be formulated by

$$L(x, \hat{\mu}) \leq \bar{z}$$

Where $L(x, \hat{\mu}) = f(x) + \hat{\mu}^T g(x)$, is a convex Lagrangian underestimating-relaxation (Nowak, 2005).

Some other valid cuts

a) Consider an MINLP problem $min\{f(x) \mid g(x) \leq 0\}$, $f: \mathbb{R}^n \mapsto \mathbb{R}$ $g: \mathbb{R}^n \mapsto \mathbb{R}^m$ and its Lagrangian function

$$L(x, \mu) = f(x) + \mu^T g(x)$$

where $\mu \in \mathbb{R}_+^m$. A lower bound ($\underline{f}$) for the problem can be computed by $\underline{f} = \min_x L(x, \mu)$ and an upper bound $\bar{f} = f(\hat{x})$ where $\hat{x}$ is a feasible point other than optimal solution. Since;

$$\underline{f} \leq L(x,\mu) \leq \bar{f} + \mu^T g(x)$$

then the following valid inequality can be defined (Nowak, 2005).

$$g_i(x) \geq -\frac{1}{\mu_i}\left(\bar{f} - \underline{f}\right)$$

b) Multiplication of two constraints also leads to a valid inequality for MINLP. Having $g(x) \leq 0$ and $h(x) \leq 0$ then, $-g(x)\,.\,h(x) \leq 0$ is a valid cut (Nowak, 2005).

2.9. Outer Approximation (OA) Algorithm

According to Conejo et al. (2006), one standard method to deal with MINLP is outer linear approximation algorithms in which in general, the original MINLP problem is first relaxed then nonlinear functions will be replaced by their linear approximation into the original problem as Master problem. Further, in each iteration the linear approximation of nonlinear functions at the current solution will be added sequentially to either relaxed or master problem as a cut to tighten the bounds on optimal value, until certain stopping criteria reached. Duran and Grossmann (1986) proposed the following outer approximation (OA) algorithm.

1- First relax the integer variables within their bounds and solve the resulting NLP problem.

2- Then a linear approximation of the nonlinear functions at the optimal solution of the relaxation will replace the nonlinear functions which leaves a MILP problem called *Master* problem.

3- Solve Master problem.

4- Fix the integer solution of the Master problem in NLP (in step.1) and solve the NLP again.

5- Again, a linearization of nonlinear functions of MINLP at the optimal solution of step.4 will be added to the Master problem.

6- Repeat steps 3 to 5 until the Master MILP problem becomes infeasible or one of the termination criteria (e.g. iteration/time limit) is satisfied.

Linear approximations are constructed by using the gradient of each nonlinear function in the NLP problem at the optimal solution of the NLP problem. The formula for the linearization of a scalar nonlinear inequality $g(x) \leq 0$ at a given point (x^0) is as follow.

$$g(x^0) + \nabla g(x^0)^T(x - x^0) \leq 0$$

The flow diagram of the OA algorithm is illustrated in figure 2.1 (Duran and Grossmann, 1986).

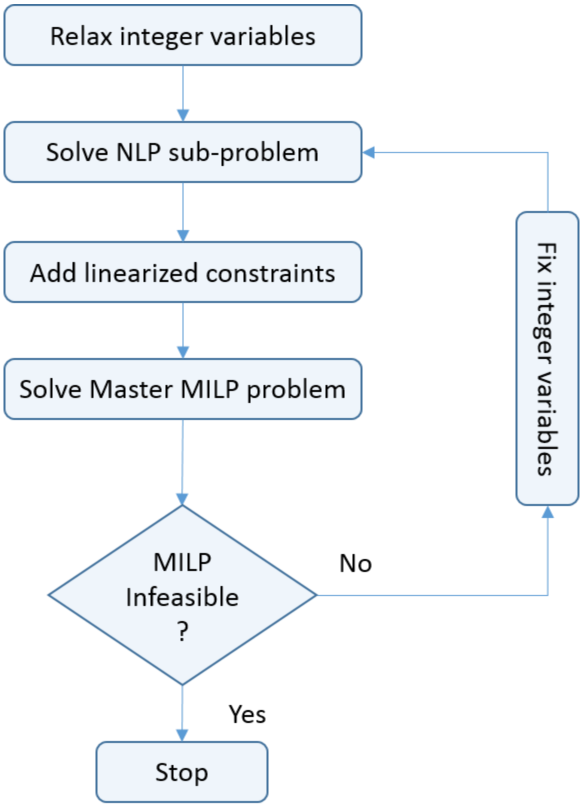

Figure 2.1: The Outer Approximation algorithm

Quesada and Grossmann (1992) noticed that for convex problems, the classic outer approximation method spend a lot of time to solve MIP, thus they proposed an outer approximation algorithm (sometimes called the LP/NLP-Based Branch-and-Bound algorithm) in which the need for restarting branch-and-bound tree search is avoided and only a single branch-and-bound tree is required. For convex problems, the algorithm will terminate if the objective of the MILP problem becomes larger than the objective of the NLP problem (in case of minimization). If the feasible region is convex then its linear approximation does not cut off any portion of solutions. Another general outer linearization algorithm is explained in Conejo et al. (2006).

Quesada and Grossmann (1992) outer approximation algorithm is as follow.

1. First relax the integer variables within their bounds and solve the resulting NLP problem.
2. Linearize nonlinear constraints at optimal solution of the relaxation and replace nonlinear constraints with the resulting linear constraints to create *Master* MILP problem.
3. Solve the master MILP problem using a branch-and-bound solver.
4. Whenever the branch-and-bound solver finds a new incumbent solution do:
    - 4.1. Solve the NLP problem by fixing the integer variables to the values in the incumbent solution.
    - 4.2. Add linearization around the optimal NLP solution as lazy constraints to the master MILP problem.
    - 4.3. Continue branch-and-bound enumeration.
5. Terminate MILP solver if the optimality gap is sufficiently small.

CHAPTER 3 METHODOLOGY AND MODELS

This chapter will review the problem to be solved and covers the methods that have been used to cope with the problem in comparison with Branch-and-Bound as an exact global solution. These methods include Outer Approximation algorithm as a general algorithm for MINLP as well as detailed description of the proposed Heuristic Branching algorithm.

Notations:

Sets:

$i \in I$: set of assets, $I = \{i\,, i = 1, \cdots, N\}$ where N is the total number of assets.

Variables:

x_i : proportion of funds to be invested in asset i

y_i : binary decision variable equal to 1 if asset i selected and 0, otherwise

Parameters:

μ_i : rate of return for asset i

Σ : covariance matrix

R : investor's minimum expected rate of return for portfolio

K : maximum number of assets to be hold in portfolio

l_i : lower bound on variable x_i if it's selected

u_i : upper bound on variable x_i if it's selected

A feasible portfolio x is called efficient if with a certain amount of risk, it provides the maximum expected return or having a certain expected return (R), it has minimum risk value. Recall the problematic from section 2.3 so called LAM as in Model 3.1

Model 3.1

$$\min_{x} \; x^T \Sigma \, x$$

S.t.

$$\sum_{i=1}^{n} \mu_i \, x_i \geq R$$

$$\sum_{i=1}^{n} x_i = 1$$

$$\sum_{i=1}^{n} y_i \leq k$$

$$l_i y_i \leq x_i \leq u_i y_i \,, \quad i = 1, \dots, n$$

$$x_i \geq 0 \qquad i = 1, \dots, n$$

$$y_i \in \{0,1\} \qquad i = 1, \dots, n$$

Since variance is always non-negative, then $x^T \Sigma \, x \geq 0$ for all x which means Σ is positive semi-definite. Model 3.1 has a convex quadratic function over a set of linear constraints. Such an MIQP model is classified as an NP-Hard problem due to combining nonlinearity and integrality in the model (Cesarone et al, 2010). Model 3.1 could be solved with CPLEX using Branch-and-Bound as a global optimal solution but restricted to small instances. In the following sections the Outer Approximation algorithm and proposed Heuristic Branching algorithm are described to overcome the problem difficulty for large size problems.

3.1. Outer Approximation Algorithm

One standard method to deal with MINLP is outer linear approximation algorithms in which in general, the original MINLP problem is first relaxed then nonlinear functions will be replaced by their linear approximation into the original problem as Master problem. Further, in each iteration the linear approximation of nonlinear functions at the current solution will be added sequentially to either relaxed or master problem as a cut to tighten the bounds on optimal value, until certain stopping criteria reached. The following is the Duran and Grossmann (1986) proposed Outer Approximation (OA) algorithm that we applied for Model 3.1. Let’s say P: iteration counter and x^p the vector of optimal solution at iteration P.

Iteration counter P $:=$ 0

Relax integer variables within their bounds in Model 3.1 ($y_i \in [0,1] \quad \forall\, i \in I$) which leaves a NLP called *Relaxed* problem.

Iteration counter P $:=$ 1

Solve *Relaxed* problem which provides a lower bound for Model 3.1. Denote the optimal solution of relaxed problem by x^1.

Iteration counter P $:=$ 2

Replace non-linear functions of Model 3.1 with linear approximation at the optimal solution x^1 which leaves a MIP called *Master* problem. Solve the Master problem and denote the solution by x^2.

Master Problem $\quad Min\ \ Z$

s.t.

$$(x^1)^T \Sigma\, x^1 + \nabla((x^1)^T \Sigma\, x^1)(x - x^1) \le Z$$

$$\sum_{i=1}^{n} \mu_i\, x_i \ge R$$

$$\sum_{i=1}^{n} x_i = 1$$

$$\sum_{i=1}^{n} y_i \le k$$

$$l_i y_i \le x_i \le u_i y_i \qquad i = 1, \dots, n$$

$$x_i \ge 0 \qquad i = 1, \dots, n$$

$$y_i \in \{0,1\} \qquad i = 1, \dots, n$$

Check the stopping criteria.

Iteration counter P $:=$ 3

Fix integer solution of the Master problem (x^{2I}) in Relaxed problem and solve it again. Denote the optimal solution of iteration $P := 3$ by x^3.

Iteration counter P $:=$ 4

Add linear approximation of non-linear functions of Model 3.1 at optimal solution x^3 to the Master problem and solve the Master problem again. Denote the solution by x^4.

$$Min\ \ Z$$

s.t.

$$\begin{aligned}
&(x^1)^T\Sigma\, x^1 + \nabla((x^1)^T\Sigma\, x^1)(x - x^1) \leq Z \\
&(x^3)^T\Sigma\, x^3 + \nabla((x^3)^T\Sigma\, x^3)(x - x^3) \leq Z \\
&\sum_{i=1}^{n} \mu_i\, x_i \geq R \\
&\sum_{i=1}^{n} x_i = 1 \\
&\sum_{i=1}^{n} y_i \leq k \\
&l_i y_i \leq x_i \leq u_i y_i \qquad i = 1, \dots, n \\
&x_i \geq 0 \qquad\qquad\quad\ \ i = 1, \dots, n \\
&y_i \in \{0,1\} \qquad\qquad i = 1, \dots, n
\end{aligned}$$

Iteration counter P $:=$ 5 to N

Repeat iteration 3 and 4 until Master problem becomes infeasible or one of the termination criteria (e.g. iteration/time limit or optimality gap) is satisfied.

The Outer Approximation algorithm will first find a lower bound in iteration.1 and an upper bound in iteration.2 then iterations 3 and 4 together will add a cut repetitively until a termination criterion like a tolerance for optimality gap is satisfied. Since OA adds cuts at each cycle it turns to slow down for large instances due to increasing size of invertible matrix.

3.2. Proposed Heuristic Branching Algorithm

To solve Markowitz portfolio optimization problem under cardinality and bounds constraints (Model 3.1), a two-phase heuristic method proposed in this research and compared with standard methods in the literature. This two-phase heuristic method decomposes the original MIQP problem into one Relaxed QP problem and one MILP problem, which both can be solved optimally in a timely manner using CPLEX.

Let us relax the MIQP problem (Model 3.1) by relaxing the binary variable y which leaves a continues convex quadratic problem (QP) called *Relaxed* problem (Model 3.2)

Model 3.2

$$\min_{x} \; x^T \Sigma \, x$$

S.t.

$$\sum_{i=1}^{n} \mu_i \, x_i \geq R \qquad (1)$$

$$\sum_{i=1}^{n} x_i = 1 \qquad (2)$$

$$\sum_{i=1}^{n} y_i \leq k \qquad (3)$$

$$l_i y_i \leq x_i \leq u_i y_i \,, \quad i = 1, \dots, n \qquad (4)$$

$$x_i \geq 0 \qquad i = 1, \dots, n$$

$$y_i \in [0,1] \qquad i = 1, \dots, n$$

The QP solvers like CPLEX can solve *Relaxed* problem to optimality very fast. The QP optimal solution x^* provides a good lower bound for the original MIQP problem. Having the QP optimal solution (x^*), the MILP will be formulated as *Master* problem (Model 3.3). This model will find an upper bound for the original MIQP problem. Denote optimal solution of *Master* problem by θ^*.

Model 3.3. *Master* problem

$$\underset{\theta}{\text{Min}}\ \underset{\theta}{\text{Max}} \sum_{i=1}^{n} |\theta_i - x_i^*|$$

S.t.

$$\sum_{i=1}^{n} \mu_i\, \theta_i \geq R$$

$$\sum_{i=1}^{n} \theta_i = 1$$

$$\sum_{i=1}^{n} y_i \leq k$$

$$l_i y_i \leq \theta_i \leq u_i y_i\,, \quad i = 1, \dots, n$$

$$\theta_i \geq 0 \quad i = 1, \dots, n$$

$$y_i \in \{0,1\} \quad i = 1, \dots, n$$

Where θ_i is a new variable expected to be as close as possible to the lower bound obtained by x_i^* for all $i = 1, \cdots, n$. The objective function of this problem is the 1-norm distance measure, seeks to minimize the maximum distance (minimizing the worst case) that can be linearized easily and leaves a MILP problem, which in turn is easy to solve to optimality using well-known solvers (CPLEX). Alternatively, the infinity-norm can be applied for objective function;

$$\|\theta - x^*\|_\infty = \max_{1\leq i\leq n} |\theta_i - x_i^*|$$

By relaxing binary variable y_i in Model (3.2) the constraint (4) will be weaken in a sense that for different values of y_i the bounds on variable x_i will change instead of being fix. In other words, the smaller the y_i, the tighter bounds on x_i will be imposed. As a consequence, more than *k* assets will be chosen which makes constraint (3) non-functional.

In order to reduce the gap between upper and lower bound (optimality gap), one of the following cuts can be added to impose binary condition on y_i in Relaxed problem (Model 3.2) which tightens the lower bound and since the upper bound tends to approach lower bound, the gap will decrease.

$$cut_i = y_i - y_i^2 = 0 \qquad \forall\, i \in I \qquad (1)$$

$$cut_i = x_i - x_i y_i = 0 \qquad \forall\, i \in I \qquad (2)$$

Both cuts (1) & (2) are non-linear and their convex linear approximations are poor. A convex approximation of equation (1) can be written as

$$\left(y_i^L + y_i^U\right) y_i - y_i^L y_i^U \geq y_i$$

Where y_i^L and y_i^U are the lower and upper bound of the variable y_i respectively. In equation (2), for bilinear term xy (product of two variables), one can replace a new variable w with bilinear terms $w = xy$. McCormick introduced the following valid inequalities which provides convex relaxation of the set of constraints $w = xy$ having bound on variables $(x^L, y^L, w^L) \leq (x, y, w) \leq (x^U, y^U, w^U)$.

$$w \geq x^L y + y^L x - x^L y^L$$

$$w \geq x^U y + y^U x - x^U y^U$$

$$w \leq x^U y + y^L x - x^U y^L$$

$$w \leq x^L y + y^U x - x^L y^U$$

These inequalities give the convex lower envelope and concave upper envelope of constraint $w = xy$. In other words, McCormick inequalities create a convex box over $w = xy$ thus tightening bound on variables, that is useful if applicable.

In case of our problem, since variable x has to be either 0 or $l \leq x \leq u$ which are fixed bounds and $y \in \{0,1\}$ is a binary variable, there is no choice for tightening bounds on variables and consequently, there would be no room for box reduction. Thus, convex linear approximations of cuts (1) & (2) are useless. One commonly used technique is to evaluate valid cuts at the optimal solution of the relaxed problem and add the most violated cut to the relaxed problem in order to tighten the bounds on the optimal value until a given optimality gap has been reached. With slightly same idea we define branching rules based on a constraint.

Let’s consider the cut_i: $y_i^R - (y_i^R)^2 = 0$, $\forall$ i to define branching rules on y_i^R, where y_i^R refers to the value of y_i in Relaxed problem. Here three strategies will be considered as bellow:

1. Strategy (1): Choose the minimum positive value of cut_i and fix the corresponding $y_i^R = 0$ in *Relaxed* problem (Model 3.2) to tighten the lower bound.
2. Strategy (2): Choose the maximum value of cut_i and fix the corresponding $y_i^R = 1$ in *Relaxed* problem (Model 3.2) to tighten the lower bound.
3. Strategy (3): Choose the minimum positive and the maximum of cut_i and fix the corresponding $y_i^R = 0$ and $y_i^R = 1$ respectively in *Relaxed* problem (Model 3.2) at the same time to tighten the lower bound.

The ***Heuristic Branching algorithm*** will be proposed based on the strategies for branching on y_i which will be applied repetitively and since the upper bound tends to approach lower bound, the gap will decrease until the optimality gap is less than a given tolerance that is the optimal solution. Here the optimality gap is defined as bellow:

$$Optimality\ Gap := ((\theta^*)^T \Sigma\, \theta^*) - ((x^*)^T \Sigma\, x^*)$$

Heuristic Branching Algorithm (HB)

```
Iteration counter $p := 1$ : Solve Relaxed problem and obtain incumbent $x_i^*$;
Iteration counter $p := 2$ : Put incumbent $x_i^*$ into Master problem and solve it. Hold incumbent $\theta_i^*$;
Update optimality gap by incumbent $x_i^*$ and $\theta_i^*$;
If optimality gap $\leq 0.001$ then STOP and return the solution, Else;
Apply one of the Strategies (1), (2) or (3) based on incumbent solution as branching rule in Relaxed problem;
Repeat;
```

Here the optimality gap ≤ 0.001 has been chosen since for smaller values in some cases the problem would be infeasible. Since HB algorithm fixes variables instead of adding cut, it does not make the problem larger in terms of convertible matrix size, hence it is computationally more efficient. In the rest of this paper, we refer to above-mentioned strategies (1), (2) and (3) by $HB\ \min(y) = 0$, $HB\ \max(y) = 1$ and HB-Mix respectively where HB stands for proposed Heuristic Branching algorithm.

CHAPTER 4 NUMERICAL RESULTS AND IMPLEMENTATION

This chapter is devoted to numerical results obtained from different methods and their comparison to distinguish the best method for different ranges of problem size. These methods include: MIQP by CPLEX as global solution compared to Mean Absolute Deviation, Gini Mean Difference, Conditional Value at Risk, Outer Approximation and proposed Heuristic Branching algorithm. A set of data has been generated randomly preserving data properties, merely to compare the performance of different methods. Note that, the same data set is used for all methods and the data itself is not the case of this research. Here is a reminder of the model and required data.

Model 4.1

$$
\begin{aligned}
&\min_{x}\ x^T \Sigma\, x \\
&\text{S.t.} \\
&\qquad \sum_{i=1}^{n} \mu_i\, x_i \geq R && && (1) \\
&\qquad \sum_{i=1}^{n} x_i = 1 && && (2) \\
&\qquad \sum_{i=1}^{n} y_i \leq k && && (3) \\
&\qquad l_i y_i \leq x_i \leq u_i y_i\,, && i = 1, \dots, n && (4) \\
&\qquad x_i \geq 0 && i = 1, \dots, n && \\
&\qquad y_i \in \{0,1\} && i = 1, \dots, n &&
\end{aligned}
$$

4.1. Numerical Results and Comparison

The required data to be generated randomly are asset's rate of return $\mu_i \in [-1,1]$ and covariance matrix Σ which is symmetric and positive semi-definite. A set of random data samples with different sizes has been generated for μ_i and Σ preserving their properties. All data samples have been generated using random continuous uniform function in MATLAB, since the assets distribution function for security market is not known in priori also it does not follow any specific distribution function. Thus, in order to maintain data randomness with equal chance of existence the uniform function adopted to generate data samples. The probability density function of the continuous uniform function is:

$$f(x) = \begin{Bmatrix} \frac{1}{b-a} \quad for \ \ a \le x \le b \ \ and \ \ 0\,, \quad otherwise \end{Bmatrix}$$

Where a and b, are minimum and maximum values of random variable x. In uniform distribution, the probability of random variables within intervals of fixed length $[a, b]$ is equal and independent of the interval location. If data are uniformly random distributed, then there is no meaningful pattern among pairs of variables. The parameters $R = 0.1$, $K = 0.6(N)$, $l_i = 0.05$ (means 5% of total fund) and $u_i = 0.4$ (means 40% of total fund) has been chosen based on rule of thumb. These parameters are usually determined by investors decision. Test problems are chosen arbitrarily.

The results are presented in the following tables. All algorithms ran and results obtained using AIMMS (4.19.4) employing CPLEX 12.6 and running under machine:

- DELL Inspiron 1520
- RAM Memory: 2.5 GB
- OS: Windows 7 Ultimate (service pack.1) 32-bit
- CPU: Intel (R) Core (TM) 2 Due CPU T7250 @ 2.00 GHz

The following tables illustrate the objective function value for different methods and sizes based on measuring $\boldsymbol{x^T \Sigma\, x}$ (in which *x* is the solution of each method) to be able to compare the methods. The MIQP in the following tables refers to the original MIQP model (Model 4.1) solved by CPLEX as an exact method which produces global solutions. The rest of methods namely MAD, GMD, CVaR, OA and HB refers to Mean Absolute Deviation, Gini Mean Difference, Conditional Value at Risk, Outer Approximation and proposed Heuristic Branching algorithm respectively. For HB algorithm, we refer to branching strategies (1), (2) and (3) by $HB\min(y) = 0$, $HB\max(y) = 1$ and *HB*-Mix respectively. It has to be mention that to run OA algorithm, a special module in AIMMS named AOA has been used.

Table 4.1: Comparison of objective function value for data samples of size 10

	Objective Function Value							
Data Sample (Size: 10)	**MAD**	**GMD**	**CVaR**	**MIQP**	**OA**	**HB Min(y)=0**	**HB Max(y)=1**	**HB Mix**
S1	0.0667	0.0687	0.067	0.0554	0.0554	0.0554	0.0554	0.0554
S2	0.0691	0.0859	0.072	0.0647	0.0647	0.0649	0.0649	0.0649
S3	0.0226	0.0447	0.031	0.0151	0.0151	0.0156	0.0156	0.0156
S4	0.0363	0.0457	0.039	0.0298	0.0298	0.0299	0.0299	0.0299
S5	0.0472	0.0454	0.045	0.0419	0.0419	0.0419	0.0419	0.0419
S6	0.043	0.0472	0.044	0.0365	0.0365	0.0366	0.0366	0.0366
S7	0.0203	0.0486	0.03	0.0171	0.0171	0.0183	0.0183	0.0183
S8	0.0354	0.0487	0.038	0.0318	0.0318	0.0318	0.0331	0.0318
S9	0.0641	0.0689	0.066	0.06	0.06	0.06	0.0603	0.0603
S10	0.0274	0.0371	0.03	0.0241	0.0241	0.0241	0.0277	0.0241
Average	0.0432	0.0541	0.0462	0.0376	0.0376	0.0379	0.0384	0.0379

Table 4.2: Comparison of objective function value for data samples of size 50

	Objective Function Value							
Data Sample (Size: 50)	**MAD**	**GMD**	**CVaR**	**MIQP**	**OA**	**HB Min(y)=0**	**HB Max(y)=1**	**HB Mix**
S1	0.0032	0.0142	0.0035	0.0021	0.0021	0.0024	0.0025	0.0023
S2	0.003	0.0123	0.008	0.0019	0.0019	0.002	0.0025	0.0022
S3	0.0037	0.0139	0.0076	0.0031	0.0031	0.0033	0.0035	0.0035
S4	0.0149	0.027	0.0172	0.0137	0.0137	0.0139	0.0139	0.0139
S5	0.0025	0.021	0.0035	0.0024	0.0024	0.0025	0.0025	0.0025
S6	0.0037	0.0148	0.0047	0.0021	0.0021	0.0024	0.0026	0.0026
S7	0.0025	0.018	0.0073	0.0021	0.0021	0.0023	0.0025	0.0025
S8	0.0288	0.0468	0.0331	0.0229	0.0229	0.0231	0.023	0.0231
S9	0.0062	0.0244	0.0085	0.0042	0.0042	0.0043	0.0049	0.0047
S10	0.0015	0.0117	0.0092	0.0012	0.0012	0.0016	0.002	0.0014
Average	0.0070	0.0204	0.0103	0.0056	0.0056	0.0058	0.0060	0.0059

Table 4.3: Comparison of objective function value for data samples of size 100

	Objective Function Value							
Data Sample (Size: 100)	**MAD**	**GMD**	**CVaR**	**MIQP**	**OA**	**HB Min(y)=0**	**HB Max(y)=1**	**HB Mix**
S1	0.0004	0.0171	0.0008	0.0003	0.0004	0.0008	0.0014	0.0012
S2	0.0008	0.0168	0.0018	0.0007	0.0007	0.001	0.0015	0.0011
S3	0.0019	0.0208	0.0021	0.0014	0.0014	0.0015	0.0017	0.0018
S4	0.0005	0.1289	0.0011	0.0004	0.0006	0.0008	0.0007	0.0012
S5	0.0003	0.1348	0.0007	0.0003	0.0005	0.0007	0.0013	0.0014
S6	0.005	0.0151	0.0061	0.0031	0.0031	0.0034	0.0033	0.0034
S7	0.0016	0.0938	0.002	0.0011	0.0013	0.0016	0.0022	0.0017
S8	0.0021	0.0089	0.0023	0.0016	0.0016	0.0017	0.0018	0.0019
S9	0.0005	0.0169	0.0007	0.0004	0.0006	0.0007	0.0013	0.0015
S10	0.0008	0.0136	0.0012	0.0006	0.0006	0.0011	0.001	0.0012
Average	0.0014	0.0467	0.0019	0.0010	0.0011	0.0014	0.0016	0.0016

According to Tables 4.1, 4.2 and 4.3 based on the average value, the methods MAD, GMD, CVaR, HB max(y)=1 and HB-Mix are totally dominated by MIQP, OA and HB min(y)=0. Thus, dominated methods will be removed form comparison for larger instances and only superior methods will remain for further comparison as illustrated in Tables 4.4, 4.5 and 4.6 bellow. In addition, the three superior methods competing closely which shows the precision of the solutions.

The results presented in Tables 4.1, 4.2 and 4.3 for problems of size 10, 50 and 100 are optimal values but the results in tables 4.4, 4.5 and 4.6 for problems of size 1000, 1500 and 2000 respectively, are objective function value obtained under time budget of 1800 seconds since for such instances solving time even exceeds one day. For problems of size 2000, MIQP and OA ran out of memory or does not produced any feasible solution within time budget of 1800 seconds which is denoted by NA (Not Applicable) but HB min(y)=0 performs efficiently which demonstrate overperformance of HB min(y)=0 in terms of computational time for large size problems.

Table 4.4: Comparison of objective function value for data samples of size 1000 up to 1800 Sec

	Objective Function Value		
Data Sample (Size: 1000)	**MIQP**	**OA**	**HB Min(y)=0**
S1	9.809	11.349	14.159
S2	39.026	11.671	14.495
S3	37.542	11.296	14.594
S4	9.842	11.187	14.197
S5	86.443	11.044	14.312
S6	9.738	11.841	14.880
S7	10.668	11.095	13.636
S8	10.294	11.326	14.800
S9	52.231	11.522	14.549
S10	9.503	11.152	14.520
Average	27.510	11.348	14.414

Table 4.5: Comparison of objective function value for data samples of size 1500 up to 1800 Sec

	Objective Function Value		
Data Sample (Size: 1500)	**MIQP**	**OA**	**HB Min(y)=0**
S1	15.491	18.567	23.106
S2	NA	17.64	22.315
S3	15.56	18.421	23.36
S4	15.491	18.57	23.106
S5	15.639	17.068	22.861
S6	15.676	17.803	21.232
S7	114.968	17.93	22.83
S8	15.169	18.018	22.34
S9	15.835	18.714	20.219
S10	111.556	17.059	22.203
Average	37.265	17.979	22.357

Table 4.6: Comparison of objective function value for data samples of size 2000 up to 1800 Sec

	Objective Function Value		
Data Sample (Size: 2000)	**MIQP**	**OA**	**HB Min(y)=0**
S1	NA	NA	31.956
S2	NA	NA	28.685
S3	NA	NA	29.781
S4	NA	NA	31.494
S5	NA	NA	29.598
S6	NA	NA	30.818
S7	NA	NA	29.748
S8	NA	NA	30.834
S9	NA	NA	29.75
S10	NA	NA	28.915
Average	---	---	30.158

Table 4.7: Comparison of computational time for data samples of size 10

	Computational Time (Seconds)							
Data Sample (Size: 10)	**MAD**	**GMD**	**CVaR**	**MIQP**	**OA**	**HB Min(y)=0**	**HB Max(y)=1**	**HB Mix**
S1	0.20	0.06	0.35	0.33	0.33	0.53	0.27	2.81
S2	0.17	0.03	0.29	0.31	0.30	0.31	0.08	0.45
S3	0.28	0.33	0.52	0.70	0.53	0.26	0.26	2.34
S4	0.56	0.06	0.61	0.70	0.55	0.27	0.26	0.81
S5	0.33	0.02	0.33	0.50	0.39	0.02	0.27	0.72
S6	0.30	0.06	0.30	0.36	0.30	0.02	0.02	0.64
S7	0.61	0.08	0.41	0.34	0.27	0.02	0.30	2.23
S8	0.38	0.06	0.43	0.48	0.30	0.26	0.02	1.53
S9	0.27	0.05	0.28	0.41	0.28	0.27	0.02	0.61
S10	0.48	0.05	0.59	0.79	0.35	0.26	0.50	2.53
Average	0.36	0.08	0.41	0.49	0.36	0.22	0.20	1.47

Table 4.8: Comparison of computational time for data samples of size 50

	Computational Time (Seconds)							
Data Sample (Size: 50)	**MAD**	**GMD**	**CVaR**	**MIQP**	**OA**	**HB Min(y)=0**	**HB Max(y)=1**	**HB Mix**
S1	1.17	492.28	3.50	0.56	2356.94	0.80	1.12	0.81
S2	0.76	61.93	3.11	0.38	53.00	1.09	1.08	0.81
S3	1.73	610.65	3.55	0.38	748.00	0.27	0.85	0.28
S4	0.39	64.00	2.47	0.27	0.31	0.01	0.01	0.05
S5	0.66	38.94	2.89	0.58	1.10	0.01	0.01	0.05
S6	0.33	572.82	2.45	0.43	133.29	0.01	0.01	0.06
S7	0.41	512.29	2.87	0.36	152.54	0.80	0.80	14.23
S8	0.30	68.00	2.62	0.28	0.53	0.30	0.53	2.48
S9	0.37	200.00	2.80	0.68	13.00	0.55	1.10	8.39
S10	0.81	367.93	3.23	0.87	1200.00	1.37	1.15	1.19
Average	0.69	298.88	2.95	0.48	465.87	0.52	0.67	2.84

Table 4.9: Comparison of computational time for data samples of size 100

	Computational Time (Seconds) ≤ 1800							
Data Sample (Size: 100)	**MAD**	**GMD**	**CVaR**	**MIQP**	**OA**	**HB Min(y)=0**	**HB Max(y)=1**	**HB Mix**
S1	1463.00	1667.46	1128.20	550.00	1800	23.14	5.75	5.18
S2	9.45	1361.65	115.38	3.50	1800	4.24	2.75	2.04
S3	6.90	2080.45	106.43	3.70	1800	3.56	3.21	2.04
S4	127.00	31.22	117.40	23.62	1800	14.80	2.85	3.45
S5	11.50	42.15	45.74	6.63	1800	15.00	4.00	3.90
S6	1.00	1901.65	33.56	1.50	179	3.00	2.20	2.14
S7	12.51	21.29	74.20	1.85	1800	5.65	2.65	2.46
S8	1.30	1005.00	88.71	1.95	1800	5.00	2.67	2.70
S9	250.00	932.00	164.70	27.35	1800	12.00	3.13	3.13
S10	91.56	18.38	89.61	9.00	1800	9.61	2.95	2.42
Average	197.42	906.13	196.39	62.91	1637.9	9.60	3.22	2.95

It has to be mentioned that for problems of size = 50 the OA algorithm will obtain the same objective values as MIQP for each sample in up to 30 seconds. In addition, for size = 100 the OA algorithm in 60 seconds will produce slightly similar results as Table 4.10 bellow.

Table 4.10: OA results in 60 seconds

	Objective function value		
Data Sample (Size: 100)	**MIQP**	**OA**	**HB Min(y)=0**
S1	0.0003	0.0007	0.0008
S2	0.0007	0.0008	0.001
S3	0.0014	0.0016	0.0015
S4	0.0004	0.0007	0.0008
S5	0.0003	0.0008	0.0007
S6	0.0031	0.0031	0.0034
S7	0.0011	0.0014	0.0016
S8	0.0016	0.0017	0.0017
S9	0.0004	0.0008	0.0007
S10	0.0006	0.0008	0.0011
Average	0.0010	0.0012	0.0014

It is worthwhile to notice that the computational time of MAD model exceeds the quadratic model for most of data samples that weakens the claim that MAD runs faster since it is linear. The proposed heuristic branching runs faster in average rather than MAD, OA and MIQP models. Moreover, GMD and CVaR turns to be computationally too slow.

Figure 4.1 depict the graph of objective values trend for MIQP, OA and $HB\ \min(y) = 0$ as the superior methods for different instances. From the figure 4.1 it is obvious that from size 100 and above, MIQP start to deviate from OA and HB and it produces higher risk.

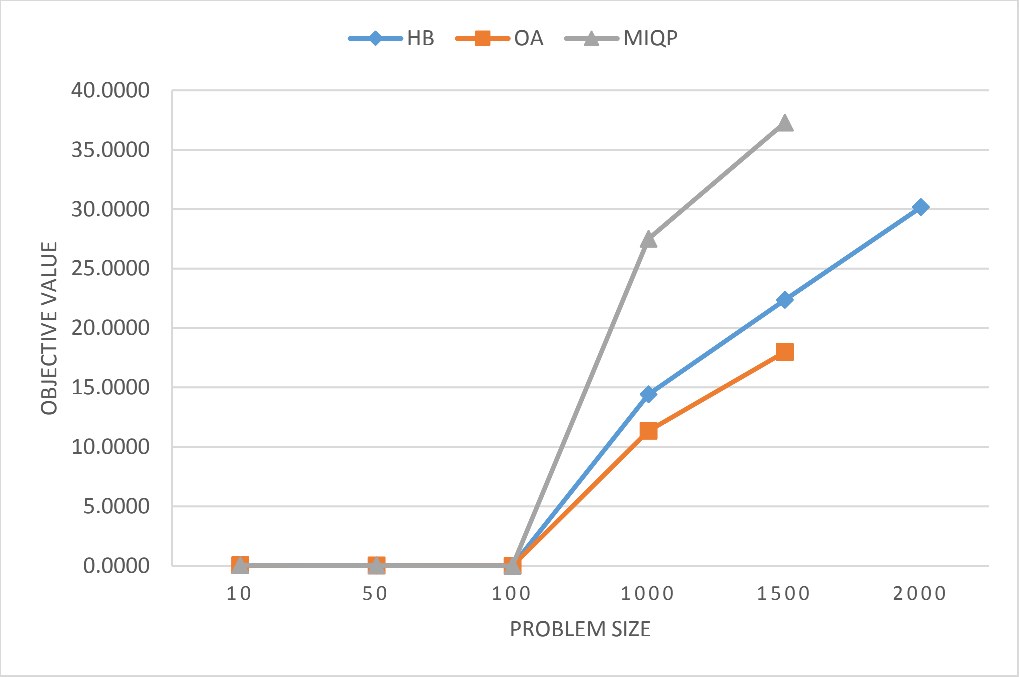

Figure 4.1: Objective value trend

In order to verify whether the complexity of the problem arise from correlations among variables and its effect on the problem, one can replace the covariance matrix with identity matrix ($\Sigma = I$) in which the variances are all equal and correlations among all pairs of variables are zero that is variables are assumed to be independent thus the effect of correlations among variables are canceled out. Table 4.11 shows results of objective value by replacing covariance matrix with identity matrix for three competing methods with time budget up to 1800 seconds.

Table 4.11: Objective value based on matrix $\Sigma = I$

	Objective Function Value		
Data Sample Size	**MIQP**	**OA**	**HB Min(y)=0**
10	0.1667	0.1667	0.1667
1000	0.05	0.05	0.05

Based on Table 4.11 the objective value for all three mentioned methods in a given time limit is equal but the computational time is still a difficulty. As a result, the degree of correlation has influence on the quality of the solution of different algorithm but not necessarily on the efficiency in term of computational time. Computational time is directly related to the complexity of covariance matrix structure and problem size, since for larger instances the inversion of the basis matrix is more difficult. It has to be mentioned that the covariance only measures the linear relationship among variables not the nonlinear which makes the problem more complicated.

The effect of parameters $\boldsymbol{K}$ and $\boldsymbol{R}$

Recall Model 4.1: It is obvious that $k \neq 0$, otherwise it forces all $y_i = 0$ hence all $x_i = 0$ which does not satisfy constraints (1) and (2) hence problem would be infeasible. For $k = 1$ only one x_i can take value and if $u_i < 1$ then there would be conflict between constraints (2) and (4) thus problem will be infeasible, however if $u_i = 1$ and there exist any $\mu_i \geq R$ then the problem has the simplest form to obtain the solution that is among all $\mu_i \geq R$ select the corresponding x_i with the smallest covariance value. For $k \geq 2$, it depends on if u_i is big enough to let constraints (1) and (2) to be satisfied. Generally, if *k* is small in relation to relatively big *R*, the problem more likely would be infeasible due to unsatisfied constraints. From another point of view, if *k* to be relatively small, then the number of assets in portfolio will be restricted that is in contrary with the concept of portfolio diversification which in turn increases the unsystematic risk. On the other side, larger *k* provides more room for well-diversification. However, a well-diversified portfolio does not necessarily consist of *k* assets but less. To evaluate the effect of small *k* let's take $k = 0.1(N)$ where *N* is the total number of assets and keep the rest of parameters unchanged. Tables 4.12 and 4.13 presents the objective values and computational times respectively for problems with size $N = 100$ and given *k =10* for three superior methods. Recalling table 4.9 in comparison with table 4.13, OA and specially MIQP turns too slow when *k* is small but still proposed *HB* algorithm runs very swiftly.

Table 4.12: Objective values based on small *k*

	Objective Function Value		
Data Sample (Size: 100)	**MIQP**	**OA**	**HB** **Min(y)=0**
S1	0.0009	0.0012	0.0022
S2	0.0021	0.0025	0.0029
S3	0.0028	0.0028	0.0040
S4	0.0012	0.0022	0.0025
S5	0.0018	0.0027	0.0037
S6	0.0040	0.0040	0.0045
S7	0.0028	0.0028	0.0030
S8	0.0022	0.0022	0.0028
S9	0.0017	0.0029	0.0029
S10	0.0018	0.0018	0.0023
Average	0.0021	0.0025	0.0031

Table 4.13: Computational time based on small *k*

	Solving Time ≤ 1800 Sec		
Data Sample (Size: 100)	**MIQP**	**OA**	**HB** **Min(y)=0**
S1	1800	1800	12.81
S2	1800	1800	12.15
S3	1545.4	1800	10.11
S4	1800	1800	13.03
S5	1800	1800	13.84
S6	10.25	1198.84	10.75
S7	1800	1800	11.82
S8	39.94	1800	10.23
S9	1800	1800	14.31
S10	1800	1800	13.23
Average	1419.56	1739.88	12.23

To investigate the effect of small K in relation to big R, to see whether the problem would be infeasible or not, consider $R = 0.2$, $K = 0.1(N)$, $l_i = 0.05$ and $u_i = 0.4$ for $N = 100$.

Table 4.14: Objective values based on small *k* and big *R*

	Objective Function Value		
Data Sample (Size: 100)	**MIQP**	**OA**	**HB** **Min(y)=0**
S1	0.0077	0.0077	0.0084
S2	0.0352	0.0352	0.0355
S3	0.0534	0.0534	0.0542
S4	0.0078	0.0078	0.0085
S5	0.0516	0.0516	0.0520
S6	NA	NA	NA
S7	0.0673	0.0673	0.0674
S8	NA	NA	NA
S9	0.0135	0.0135	0.0140
S10	0.0218	0.0218	0.0219
Average	0.0323	0.0323	0.0327

According to table 4.14 the samples S6 and S8 are shown by NA which are infeasible due to unsatisfied constraint (1) and it proves that the chance of problem infeasibility increases when *K* is smaller in relation to bigger *R*.

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS

This chapter is mainly to summarize the results obtained in chapter 4 in a concise way and to classify presented methods based on problem size. According to the results in tables 4.1, 4.2 and 4.3 for the proposed Heuristic Branching algorithm, Strategy (1) defeat strategies (2) and (3) thus it will be selected as superior branching rule for the HB algorithm.

Tables 4.1, 4.2 and 4.3 empirically proves the competency among discussed methods. Tables 4.4, 4.5 and 4.6 reveals that the MIQP is only applicable for small size problem with a few hundred variables. However, the problematic is to deal with large size problems say with one thousand variables. In such a case, OA algorithm showed overperformance over the other methods as presented in table 4.4 and 4.5 for medium size problems up to two thousand variables. The heuristic branching algorithm demonstrated its efficiency especially for large size problem as depicted in table 4.6 for problem sizes around two thousand variables and more.

To sum up, portfolio optimization problem with cardinality constraint is a very challenging problem and its complexity stems from the structure of the covariance matrix as the risk measure and the size of invertible matrix as it increases, which makes the problem more difficult it terms of computational time. Exact methods like Branch & Bound that enumerates all solutions in branching tree are too expensive to deal with such an NP-Hard problem. On the other hand, algorithms that make use of adding valid cuts to tighten the optimality gap, turn to be too slow as the size of problem becomes larger. Therefore, developing algorithm that benefits from smart branching rules and conditions will be fruitful especially for big size problems. Here the logic is to trade-off between the solution precision and computational time.

BIBLIOGRAPHY

Adjiman, C.S., Schweiger, C.A. & Floudas, C.A. (1991). Non-linear and Mixed-Integer Optimization in chemical process network systems. DIMACS series in Discrete Mathematics and Theoretical Computer Science.

Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J. & Mahajan, A. (2012). Mixed-Integer Nonlinear Optimization. Mathematics and Computer Science Division, Argonne National Laboratory.

Bertismas, D. & Shioda, R. (2009). Algorithm for cardinality constrained quadratic optimization. Comput Optim Appl, 43, 1-12.

Bienstock, D. (1996). Computational study of a family of mixed-integer quadratic programming problems. Math program, 74, 121-140.

Brito, R.P. & Vicente, L.N. (2008). Efficient Cardinality/Mean-Variance Portfolios. Department of Mathematics, University of Coimbra, 3001-454 Coimbra, Portugal.

Brigham, E.F. & Ehrhardt, M.C. (2008). Financial Management; Theory and practice. 12e, Thomson, south-western.

Brigham, E.F. & Houston, J.F. (2007). Fundamentals of Financial Management. 11e, Thomson, south-western.

Burer, S. & Letchfordy, A.N. (2012). Non-Convex Mixed-Integer Nonlinear Programming: A Survey. Surveys in Operations Research and Management Science, 17, 97–106.

Cesarone, F., Scozzari, A. & Tardella, F. (2010). Portfolio selection problems in practice: a comparison between linear and quadratic optimization models.

Chang, T.J., Meade, N., Beasley, J.E. & Sharaiha, Y.M. (2000). Heuristics for cardinality constrained portfolio optimization. Computers & Operations Research, 27 (2000), 1271-1302.

Chen, D.S., Batson, R.G. & Dang, Y. (2010). Applied Integer Programming: Modeling and Solution. John Wiley & Sons Inc.

Conejo, A.J., Castillo, E., Mínguez, R. & García-Bertrand, R. (2006). Decomposition Techniques in Mathematical Programming. Springer publication.

Cornuejols, G. & Tutuncu, R. (2006). Optimization Methods in Finance. Cambridge, UK: Cambridge University Press.

Cui, T., Cheng, S. & Bai, R. (2014). A Combinatorial Algorithm for The Cardinality Constrained Portfolio Optimization Problem. IEEE Congress on Evolutionary Computation.

Duran, M.A. & Grossmann, I.E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming, 36, 307–339.

Junger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., Rinaldi, G. & Wolsey, L. (2010). 50 Years of Integer Programming 1958–2008. Berlin: Springer. Doi: 10.1007/978-3-540-68279-0, ISBN 978-3-540-68274-5.

Konno, H. & Yamazaki, H. (1991). Mean absolute deviation portfolio optimization model and its applications to Tokyo stock market. Management Science, 37, 519-529.

Lee, J. & Leyffer, S. (2012). Mixed Integer Nonlinear Programming. The IMA Volumes in Mathematics and its Applications, 154, Springer. doi: 10.1007/978-1-4614-1927-3.

LI, D., Sun, X.L. & Wang, J. (2006). Optimal lot solution to cardinality constrained Mean–Variance formulation for portfolio selection. Mathematical Finance, 16 (1), 83–101.

Locatelli, M. & Schoen, F. (2013). Global Optimization; Theory, Algorithms and Applications. SIAM.

Mansini, R., Ogryczak, W. & Speranza, M.G. (2015). Linear and Mixed Integer Programming for Portfolio Optimization. Euro Advanced Tutorials on Operational Research, Springer, Switzerland.

Maringer, D. (2005). Portfolio Management with Heuristic Optimization. Springer.

Moyer, R.C., McGuigan, J.R. & Kretlow, W.J. (2006). Contemporary Financial Management, 10e. South-Western, Thomson.

Murray, W. & Shek, H. (2012). A local relaxation method for the cardinality constrained portfolio optimization problem. Comput Optim Appl, doi: 10.1007/s10589-012-9471-1.

Nemhauser, G. & Wolsey, L. (1999). Integer and Combinatorial Optimization. John Wiley & Sons Inc.

Nowak, I. (2005). Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming, International Series of Numerical Mathematics, 152, Birkhäuser Verlag, ISBN 3-7643-7238-9.

Prigent, J.L. (2007). Portfolio Optimization and Performance Analysis. Chapman & Hall/CRC Financial Mathematics Series, Taylor & Francis Group.

Quesada, I. & Grossmann, I.E. (1992). An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. Computers and Chemical Engineering, 16, 937-947.

Rudolf, M., Wolter, H.J. & Zimmermann, H. (1999). A linear model for tracking error minimization. Journal of Banking and Finance, 23, 85-103.

Sadjadi, S.J., Gharakhani, M. & Safari, E. (2012). Robust optimization framework for cardinality constrained portfolio problem. Applied Soft Computing, 12, 91–99.

Stoyan, S.J. (2009). Advances in Portfolio Selection Under Discrete Choice Constraints: A Mixed Integer Programming Approach and Heuristics. A thesis submitted for Doctor of Philosophy, University of Toronto.

Tapiero, C. (2004). Risk and Financial Management: Mathematical and Computational Methods. John Wiley & Sons Ltd.

Wolsey, L.A. (1998). Integer Programming. John Wiley & Sons, Inc.