Evolutionary Approaches for Portfolio Optimization

Khin Thein Lwin, BSc (Hons)

Department of Computer Science

University of Nottingham

A thesis submitted for the degree of

Doctor of Philosophy

March 2015

Abstract

Portfolio optimization involves the optimal assignment of limited capital to different available financial assets to achieve a reasonable trade-off between profit and risk objectives. Markowitz's mean variance (MV) model is widely regarded as the foundation of modern portfolio theory and provides a quantitative framework for portfolio optimization problems. In real market, investors commonly face real-world trading restrictions and it requires that the constructed portfolios have to meet trading constraints. When additional constraints are added to the basic MV model, the problem thus becomes more complex and the exact optimization approaches run into difficulties to deliver solutions within reasonable time for large problem size. By introducing the cardinality constraint alone already transformed the classic quadratic optimization model into a mixed-integer quadratic programming problem which is an NP-hard problem. Evolutionary algorithms, a class of metaheuristics, are one of the known alternatives for optimization problems that are too complex to be solved using deterministic techniques.

This thesis focuses on single-period portfolio optimization problems with practical trading constraints and two different risk measures. Four hybrid evolutionary algorithms are presented to efficiently solve these problems with gradually more complex real world constraints. In the first part of the thesis, the mean variance portfolio model is investigated by taking into account real-world constraints. A hybrid evolutionary algorithm (PBILDE) for portfolio optimization with cardinality and quantity constraints is presented. The proposed PBILDE is able to achieve a strong synergetic effect through hybridization

of PBIL and DE. A partially guided mutation and an elitist update strategy are proposed in order to promote the efficient convergence of PBILDE. Its effectiveness is evaluated and compared with other existing algorithms over a number of datasets. A multi-objective scatter search with archive (MOSSwA) algorithm for portfolio optimization with cardinality, quantity and pre-assignment constraints is then presented. New subset generations and solution combination methods are proposed to generate efficient and diverse portfolios. A learning-guided multi-objective evolutionary (MODEwAwL) algorithm for the portfolio optimization problems with cardinality, quantity, pre-assignment and round lot constraints is presented. A learning mechanism is introduced in order to extract important features from the set of elite solutions. Problem-specific selection heuristics are introduced in order to identify high-quality solutions with a reduced computational cost. An efficient and effective candidate generation scheme utilizing a learning mechanism, problem specific heuristics and effective direction-based search methods is proposed to guide the search towards the promising regions of the search space.

In the second part of the thesis, an alternative risk measure, VaR, is considered. A non-parametric mean-VaR model with six practical trading constraints is investigated. A multi-objective evolutionary algorithm with guided learning (MODE-GL) is presented for the mean-VaR model. Two different variants of DE mutation schemes in the solution generation scheme are proposed in order to promote the exploration of the search towards the least crowded region of the solution space. Experimental results using historical daily financial market data from S &P 100 and S & P 500 indices are presented. When the cardinality constraints are considered, incorporating a learning mechanism significantly promotes the efficient convergence of the search.

List of Publications

The following work was published/submitted for publication as a result of the investigations performed in the course of this thesis.

- K. Lwin and R. Qu (2013). A Hybrid Algorithm for Constrained Portfolio Selection Problem. *Applied Intelligence*, 39(2):251-266.

- K. Lwin and R. Qu (2013). Multi-objective Scatter Search with External Archive for Portfolio Optimization. *The 5^{th} International Conference on Evolutionary Computation Theory and Applications (ECTA2013)*, pages. 111-119, 20-22 September, Algrave, Portugal, 2013.

- K. Lwin, R. Qu and G. Kendall (2014). A Learning-guided Multi-objective Evolutionary Algorithm for Constrained Portfolio Optimization. *Applied Soft Computing*, 24(0): 757-772.

- K. Lwin, R. Qu and B. Maccarthy (2014). Mean-VaR Portfolio Optimization: A Non-parametric Approach, under review at *European Journal of Operational Research*, 2014.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Rong Qu, for her invaluable guidance, continuous support and encouragement throughout this research. I would also like to thank Prof. Bartholomew Maccarthy from Business School for his great suggestions and guidance for the work in Chapter 7.

I would like to thank my financial sponsor, the University of Nottingham (UON), for providing the funding to undertake this research. My PhD would not have been possible without the two scholarships (PhD Studentship & International Research Excellence Scholarship) funded by the School of Computer Science and the University of Nottingham.

I am also very grateful to the members of my thesis committee - Professor Edward Tsang[a] and Dr. Jason Atkin[b] for providing me with valuable comments on this thesis.

I would also like to express my heartfelt gratitude to my former undergraduate supervisor, Dr. Natasha Alechina, who I enormously admire. Ever since I met her, she has always been motivating and supportive mentor and teacher who would listen to my goals and provide me all information and guidances to pursue my goals. I still fondly think of my time as an undergraduate student under her supervision and I feel extremely privileged to be one of her students.

[a] Center for Computational Finance and Economic Agents, School of Computer Science and Electronic Engineering, University of Essex, UK.
[b] ASAP Research Group, School of Computer Science, University of Nottingham, UK.

I would also like to thank Dr. Hai Nguyen who becomes my best friend since the undergraduate studies in Nottingham for many years of true friendship. My gratitude is also extended to Prof. Roland Backhouse and Dr. Dario Landa-Silva for their encouragement throughout this long journey.

I would like to thank all past and present members in ASAP research group for contributing such an enjoyable and productive research environment. I would also like to take this opportunity to acknowledge a group of lovely friends who came into my life during my time in Nottingham and they have made my daily life in academia quite enjoyable. Thank you all for your genuine friendship and the wonderful time together! Special thanks goes to Dr. Grazziela Figueredo, Dr. Yijun Wang, Dr. Huanlai Xing, Dr. Ahmed Kheiri, Dr. Daniel Karapetyan, Dr. Jenna Reps, Dr. Tuan Nguyen, Ann Colin, Yagoub Shadad, Dr. Daphne Lai, Dr. Jakub Marecek, Heshan Du, Dr. Urszula Neuman, Tamanna Rahman, Dr. Ayodele Oladeji, Charlotte May, Karolina Wysoczanska, Shahriar Asta, Arturo Castillo, Ahmad Muklason, Dr. Anas Elhag and Dr. Ha Thai Duong.

Finally, and most importantly, I am forever grateful to my parents and brother for their understanding and encouragement to pursue my goals. Without their unconditional love and support, I would never have got anywhere near this stage. My brother has always been my real-life superhero who always give me the world's best advices throughout my life. Words fails me when I try to express my infinite gratitude to my supportive family. I would like to dedicate this thesis to my parents and brother for their constant support and unconditional love. This thesis is also dedicated to the loving memory of my late aunty, Se Yee. I love you all dearly.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

CLA	Critical Line Algorithm
DE	Differential Evolution
DEMO	Differential Evolution for Multi-objective Optimization
EA	Evolutionary Algorithm
EDA	Estimation of Distribution Algorithm
EF	Efficient Frontier
ES	Expected Shortfall
GA	Genetic Algorithm
GD	Generational Distance
HC	Hill Climbing
HV	Hypervolume
IBEA	Indicator-based Evolutionary Algorithm
IGD	Inverted Generational Distance
MA	Memetic Algorithm
MIQP	Mixed Integer Quadratic Programming
MODE	Multi-objective Differential Evolution
MOEA	Multi-objective Evolutionary Algorithm
MOEA/D	Multi-objective Evolutionary Algorithm with Decomposition
MOP	Multi-objective Optimization Problems
MV	Mean-Variance
NSGA	Nondominated Sorting Genetic Algorithm
NSGA-II	Elitist Nondominated Sorting Genetic Algorithm
PAES	Pareto Archived Evolutionary Strategy
PBIL	Population Based Incremental Learning

PESA	Pareto Envelope-based Selection Algorithm
PESA-II	Improved Pareto Envelope-based Selection Algorithm
PSO	Particle Swarm Optimization
PSP	Portfolio Selection Problem
QP	Quadratic Programming
SA	Simulated Annealing
SOEA	Single Objective Evolutionary Algorithm
SPEA	Strength Pareto Evolutionary Algorithm
SPEA2	Improved Strength Pareto Evolutionary Algorithm
SS	Scatter Search
TS	Tabu Search
UCEF	Unconstrained Efficient Frontier
VaR	Value-at-Risk
VEGA	Vector Evaluated Genetic Algorithm

Roman Symbols

$\mathcal{A}_{size}$	Archive size
$\mathcal{B}$	Reference set size
$\mathcal{CR}$	Crossover probability
$\mathcal{F}$	Amplification factor
$\mathcal{K}$	Number of assets in a portfolio
$\mathcal{LR}$	Learning rate
$\mathcal{MP}$	Mutation Probability
$\mathcal{MR}$	Mutation Rate
$\mathcal{M}$	Number of asset class
$\mathcal{NP}$	Number of individuals in a population
$\mathcal{N}$	Number of available assets
$\mathcal{N\_LR}$	Negative learning rate
$\mathcal{S}$	Subset set (sub) size

Chapter 1

Introduction

> "Great stocks are extremely hard to find. If they weren't, then everyone would own them."
>
> Philip A. Fisher

1.1 Background and Motivation

From the financial point of view, a portfolio is a collection of investments held by an individual or a financial institution. These investments can be financial assets ranging from stocks, bonds, or options to real estate. In financial markets, there exists a huge variety of asset classes in which one may invest his/her wealth. Different assets have different levels of risk. Different investors have their own attitude towards the risk. Given an extensive range of financial assets with different characteristics, the essence of the problem is to find a combination of assets that serves the best for an investor's needs.

In 1952, Markowitz addressed a fundamental question in financial decision making: How should an investor allocate his/her wealth among the possible investment choices? Markowitz introduced a parametric optimization model by proposing that investors should decide the allocation of their investments based on a trade-off between risk and return. Markowitz's mean variance (MV) model

proposes that investment returns can be represented by a weighted average of the returns of the underlying assets and risk is reflected as the variability of payoffs. Markowitz's mean variance (MV) principle (Markowitz, 1952, 1959) is considered to play an important role in the development of modern portfolio theory.

Many investment situations may make investment managers consider MV framework for wealth allocation. Based on market index historic returns, an international equity manager may need to find optimal asset allocations among international equity markets. A plan sponsor may like to find an optimal long-term investment policy for allocating among different classes such as domestic, foreign bonds and equities. A domestic equity manager may wish to find an optimal equity portfolio based on forecasts of return and estimated risk (Michaud and Michaud, 2008).

MV optimization model is useful as an asset management tool for many applications, such as (Michaud and Michaud, 2008):

- Implementing investment objectives and constraints
- Controlling the components of portfolio risk
- Implementing the asset manager's investment strategies
- Using active return information efficiently
- Embedding new information into portfolios efficiently

Moreover, the MV optimization model is flexible enough to reflect various practical trading constraints and it can thus be served as the standard optimization framework for modern asset management (Michaud and Michaud, 2008).

There are exact methods such as simplex methods (Dantzig, 1998), interior point methods (Adler et al., 1989) and quadratic programming methods (Hirschberger et al., 2010; Markowitz, 1987; Stein et al., 2008) which can be employed in order

to find the optimal solution for the basic MV model with a reasonable computational effort. However, these methods can be applied to problems satisfying certain conditions such as the objective function must be of a certain type, the constraints must be expressible in certain formats, and so on (Boyd and Vandenberghe, 2004). Without modifying and/or simplifying the problems into solvable forms, the applications of these methods are therefore limited to a certain set of problems (Maringer, 2005).

The basic MV framework for portfolio optimization assumes markets to be frictionless. In real market, investors commonly face real-world trading restrictions and it requires that the constructed portfolios have to meet trading constraints. Investors also have their own preferences and this may lead to impose further constraints in allocating capital among the assets. It is therefore needed to extend the standard model in order to reflect practical trading restrictions and investors' valuable insights.

When additional constraints are added to the basic MV model, the problem thus becomes more complex and the exact optimization approaches run into difficulties to deliver solutions within reasonable time for large problem size. By introducing the cardinality constraint alone already transformed the classic quadratic optimization model into a mixed-integer quadratic programming problem which is an NP-hard problem (Bienstock, 1996; Moral-Escudero et al., 2006; Shaw et al., 2008). As a result, this motivates the investigation of approximate algorithms such as metaheuristics (Gendreau and Potvin, 2010; Glover and Kochenberger, 2003) and hybrid meta-heuristics (Raidl, 2006; Talbi, 2002). In general, metaheuristics cannot guarantee the optimality of the solution, but they are efficient in finding the optimal or near optimal solutions in a reasonable amount of time.

Markowitz (1959) also noted that risk quantification for portfolio optimization is an open problem since it depends on the investor's needs. No one risk measure, therefore, may satisfy different needs of different investors. Many studies have been conducted to quantify the portfolio risk with different measures.

A particular class of measure which quantify possibilities of return below expected return are called downside risk measures (Harlow, 1991; Krokhmal et al., 2011). Among those downside risk measures, Value-at-Risk (VaR) (Morgan, 1996) is a popular measurement of risk widely recognized by financial regulators and investment practitioners. The portfolio optimization in the VaR context involves additional complexities since VaR is non-linear, non-convex and non-differentiable, and it exhibits multiple local extrema and discontinuities especially when real-world trading constraints are incorporated (Gaivoronski and Pflug, 2005). In fact Benati and Rizzi (2007) show that optimization of the mean-VaR portfolio problem leads to a non-convex NP-hard problem which is computationally intractable.

In the past decade, there has been an increasing interest to explore the application of evolutionary algorithms for portfolio optimization problems. Evolutionary algorithms, a class of metaheuristics, are one of the known alternatives for optimization problems that are too complex to be solved using deterministic techniques. They are independent of the types of objective function and the constraints while also being attractive for their capability to solve computationally demanding problems reliably and efficiently.

The motivation for this thesis is based on three main avenues in the literature on portfolio optimization. The first area of interest is to design hybrid evolutionary algorithms for portfolio optimization problems. In particular, we are interested in integrating selective properties of different evolutionary approaches in order to mitigate their individual weaknesses and achieve efficient convergence of the search. The second area of interest is to extend the basic model with practical trading constraints in order to better reflect the practical trading limitations. Recent review by Metaxiotis and Liagkouras (2012) shows that the cardinality and quantity constraints are the most commonly considered constraints in the literature. Therefore, we are interested in investigating the portfolio optimization models as realistic as possible by considering increasing number of practical trading constraints. The third area of interest is to adopt VaR as an alternative risk measure in place of the variance. Recent surveys by Metaxiotis and Liagk-

ouras (2012) and Ponsich et al. (2013) also show that the research in portfolio optimization in the nonparametric mean-VaR framework is still in its infancy compared to mean variance framework.

1.2 Aims and Objectives

The goal of this thesis is to provide a contribution to portfolio optimization research through the development of efficient and effective algorithms and to investigate their applications to portfolio optimization problems with additional practical trading constraints. In order to achieve this goal, the identified objectives are as follows:

- To extend the basic portfolio model as realistic as possible by considering increasing number of practical trading constraints.
- To design and investigate the ability of single objective evolutionary algorithms to deliver high-quality solutions for the constrained portfolio optimization problems.
- To design effective and efficient multi-objective evolutionary algorithms for portfolio optimization problems reflecting practical trading constraints.
- To conduct a fair performance comparison between the proposed algorithms and existing state-of-the-art evolutionary algorithms.
- To investigate an alternative industry standard risk measure for the portfolio optimization problems in order to capture the asymmetric nature of risk.

1.3 Contributions

The contributions of this thesis can be summarized as follows:

- A hybrid evolutionary algorithm (PBILDE) is developed to solve the portfolio optimization problems with cardinality and quantity constraints (see

Chapter 4). A partially guided mutation and an elitist update strategy are proposed in order to promote the efficient convergence of PBILDE. PBILDE is able to achieve a strong synergetic effect through hybridization of PBIL and DE. In most problem instances, it also outperforms other existing approaches in the literature which adopted the same mean variance model.

- A multi-objective scatter search with external archive (MOSSwA) algorithm is proposed for the first time for portfolio optimization problems with cardinality, quantity and pre-assignment constraints (see Chapter 5). MOSSwA adapts the basic scatter search template to multi-objective optimization by incorporating the concepts of Pareto dominance, crowding distance and elitism. New subset generations and solution combination methods are proposed to generate efficient and diverse portfolios. MOSSwA outperforms NSGA-II, SPEA2 and PESA-II in all five problem instances both in terms of solution quality and computational time.

- A learning-guided multi-objective evolutionary (MODEwAwL) algorithm is developed to solve the portfolio optimization problems with cardinality, quantity, pre-assignment and round lot constraints (see Chapter 6). A learning mechanism is introduced in order to extract important features from the set of elite solutions. Problem-specific selection heuristics are introduced in order to identify high-quality solutions with a reduced computational cost. An efficient and effective candidate generation scheme utilizing a learning mechanism, problem specific heuristics and effective direction-based search methods is proposed to guide the search towards the promising regions of the search space. In small problem instances, MODEwAwL is competitive to NSGA-II and SPEA2. In large problem instances, MODEwAwL achieves better performance over four existing well-known MOEAs, NSGA-II, SPEA2, PEAS-II and PAES. The computational results not only show that the quality of the generated solutions significantly improved, but also that the overall computation time can be reduced.

- Value-at-risk (VaR), an industry standard risk measure, is studied in order to reflect a realistic risk measure. The mean-VaR portfolio optimization

problem with six practical constraints is for the first time considered (see Chapter 7). A multi-objective evolutionary algorithm with guided learning (MODE-GL) is developed to solve the constrained mean-VaR portfolio optimization problems. Two different variants of DE mutation schemes in the solution generation scheme are proposed in order to promote the exploration of the search towards the least crowded region of the solution space. When the cardinality constraints are considered, incorporating a learning mechanism significantly promotes the efficient convergence of the search.

1.4 Outline

The structure of this thesis can be summarized as follows. Chapter 2 provides an introduction to the background of the thesis, through a brief overview of variants of optimization approaches for the single-period portfolio optimization models. A number of practical constraints commonly faced by investors and datasets utilized for computational analysis in this thesis are also described. Chapter 3 provides an overview of the key concepts in multi-objective optimization problems. Most well-known population-based evolutionary algorithms are reviewed and their applications are summarized.

Chapter 4 presents a hybrid algorithm for portfolio optimization problem with cardinality and quantity constraints and investigates the effectiveness of the components of the algorithm. Chapter 5 describes a multi-objective scatter search algorithm for portfolio optimization problems with three constraints. Chapter 6 presents a learning-guided multi-objective evolutionary algorithm for the mean variance portfolio optimization problems. Chapter 7 studies the Value-at-Risk (VaR) as an alternative risk measure and presents a multi-objective evolutionary algorithm with guided learning for mean-VaR portfolio optimization problems. Chapter 8 concludes with a summary and suggestions for future research directions.

Chapter 2

Portfolio Optimization

> "It's not whether you're right or wrong that's important, but how much money you make when you're right and how much you lose when you're wrong."
>
> George Soros

2.1 Introduction

Portfolio optimization plays an important decision making role in investment management. It is concerned with the optimal allocation of a limited capital among a finite number of available assets, such as stocks, bonds and derivatives, in order to gain the highest possible future return subject to a tolerance level at the end of the investment period. Mean-variance portfolio formulation (Markowitz, 1952, 1959) pioneered by Nobel Laureate Harry Markowitz has provided an influential insight into decision making concerning the capital investment in modern computational finance. Since the return of the investment is not guaranteed but approximated (i.e., expected), a variation of the return should be considered as the risk of receiving the expected return. Markowitz therefore reasoned that investors should not only be concerned with the realized returns, but also the risk associated with the asset holdings and introduced the

portfolio optimization as a mean variance optimization problem with regard to two criteria: to maximize the reward of a portfolio (measured by the mean of expected return), and to minimize the risk of the portfolio (measured by the variance of the return). In the simplest sense, a desirable portfolio is defined to be a trade-off between risk and expected return.

This chapter provides an introduction to the background of the thesis, through a review of the relevant portfolio optimization problems with different approaches. A portfolio optimization model with an alternative risk measure is also described. In addition, a number of real-world trading constraints commonly faced by investors are discussed. The detailed descriptions of the datasets used in this thesis for computational analysis are also presented.

2.2 Markowitz's Mean-Variance Model

Markowitz (1952, 1959) introduced a parametric optimization model in a mean variance framework which provides analytical solutions for an investor either trying to maximize his/her expected return for a given level of risk or trying to minimize the risk for a given level of expected return. The mean variance (MV) model assumes that the future market of the assets can be correctly reflected by the historical market of the assets. The reward (profit) of the portfolio is measured by the average expected return of those individual assets in the portfolio whereas the risk is measured by its combined total variance or standard deviation. Markowitz's mean variance model (MV model) is formulated as an optimization problem over real-valued variables with a quadratic objective function and linear constraints as follows:

$$minimize \qquad \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} w_i w_j \sigma_{ij} \tag{2.1}$$

$$subject\ to \qquad \sum_{i=1}^{\mathcal{N}} w_i \mu_i = \mathcal{R}^* \tag{2.2}$$

$$\sum_{i=1}^{\mathcal{N}} w_i = 1 \tag{2.3}$$

$$0 \leq w_i \leq 1, i = 1, \ldots, \mathcal{N} \tag{2.4}$$

where $\mathcal{N}$ is the number of available assets, μ_i is the expected return of asset i ($i = 1, \ldots, \mathcal{N}$), σ_{ij} is the covariance between assets i and j ($i = 1, \ldots, \mathcal{N}; j = 1, \ldots, \mathcal{N}$), $\mathcal{R}^*$ is the desired expected return, and w_i ($0 \leq w_i \leq 1$) is the decision variable which represents the proportion held of asset i. Eq. (2.1) minimizes the total variance (risk) associated with the portfolio whilst Eq. (2.2), the *return constraint*, ensures that the portfolio has a predetermined expected return of $\mathcal{R}^*$. Eq. (2.3) defines the *budget constraint* (all the money available should be invested) for a feasible portfolio while Eq. (2.4) requires that all investment should be *positive*, i.e., no short sales are allowed.

2.2.1 Single Objective Mean-Variance Model

An alternative form of the MV model can be formulated by introducing a risk aversion parameter $\lambda \in [0, 1]$ to form an aggregate objective function which is a weighted combination of both return and risk as follows:

$$minimize \quad \lambda \left[\sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} w_i w_j \sigma_{ij} \right] + (1 - \lambda) \left[- \sum_{i=1}^{\mathcal{N}} w_i \mu_i \right] \tag{2.5}$$

$$subject\ to \quad \sum_{i=1}^{\mathcal{N}} w_i = 1 \tag{2.6}$$

$$0 \leq w_i \leq 1, i = 1, \ldots, \mathcal{N} \tag{2.7}$$

In Eq. (2.5), when λ is zero, the model maximizes the mean expected return of the portfolio regardless of the variance (risk). On the other hand, when λ equals one, the model minimizes the risk of the portfolio regardless of the mean

expected return. As the λ value increases, the relative importance of the return decreases, and the emphasis of the risk to the investor increases, and vice versa.

2.2.2 Multi-objective Mean-Variance Model

Mean-Variance model is considered to be the first systematic treatment of investor's conflicting objectives of higher return versus lower risk. Portfolio optimization problem is intrinsically a multi-objective problem since the objective is to find portfolios amongst the $\mathcal{N}$ assets that can simultaneously satisfy the above two conflicting objectives, i.e., minimize the total variance (see Eq. (2.8)), denoting the risk associated with the portfolio, while maximizing its profits (see Eq. (2.9)). The portfolio optimization problem can therefore be restated as:

$$min \qquad f_1 = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} w_i w_j \sigma_{ij} \tag{2.8}$$

$$max \qquad f_2 = \sum_{i=1}^{\mathcal{N}} w_i \mu_i \tag{2.9}$$

$$subject\ to \quad \sum_{i=1}^{\mathcal{N}} w_i = 1 \tag{2.10}$$

$$0 \leq w_i \leq 1, i = 1, ..., \mathcal{N} \tag{2.11}$$

The standard model, single objective model and multi-objective model are three well-established approaches commonly adopted to solve the portfolio problem. Chang et al. (2000) stated that the solutions for the basic portfolio optimization problem can be achieved by either solving the classic MV model (see Eqs. (2.1) to (2.4)) varying λ or solving the combined objective model (see Eqs. (2.5) to (2.7)) varying $\mathcal{R}^*$. Which of these models to be selected depends on the goal of the optimization and on the capabilities of the available software packages. Most researchers commonly adopt the last two models when they use a heuristic approach (Metaxiotis and Liagkouras, 2012; Ponsich et al., 2013).

2.2.3 Efficient Frontier

Finance theory argues that risk and expected returns are positively related, which implies that higher returns are achievable only when investors are willing to take higher risks and vice versa, i.e. the risk cannot be reduced without decreasing the return (Weigand, 2014). In practice, different investors have different preferred trade-offs between risk and expected return. An investor who is very risk-averse will choose a safe portfolio with a low risk and a low expected return. Conversely, an investor who is less risk averse will choose a more risky portfolio with a higher expected return. Thus, the portfolio optimization problem does not prescribe a single optimal portfolio combination that both minimizes variance and maximizes expected return. Instead, the result of the portfolio optimization is generally a range of efficient portfolios.

A portfolio is said to be *efficient* (i.e., *Pareto optimal*) in the context of mean variance portfolio optimization if and only if there is no other feasible portfolio that improves at least one of the two optimization criteria without worsening the other (see Section 3.2.1). In a two-dimensional space of risk and return, a solution a is efficient if there does not exist any solution b such that b dominates a (Fonseca and Fleming, 1995). Solution a is considered to dominate solution b if and only if C_1 or C_2 holds:

$$C_1\text{:} \quad f_1(a) \leq f_1(b) \quad \wedge \quad f_2(a) > f_2(b)$$

$$C_2\text{:} \quad f_2(a) \geq f_2(b) \quad \wedge \quad f_1(a) < f_1(b)$$

The collection of these efficient portfolios forms the *efficient frontier* (i.e., *Pareto front*) that represents the best trade-offs between the return and the risk[1]. We could trace out the set of efficient portfolios by solving the model (Eqs 2.5 – 2.7) repeatedly with a different value of λ at each time. Figure 2.1 shows the efficient frontier (EF) plotted in the risk-return solution space for a 31-asset universe of Hang Seng dataset from the OR-library (see Section 2.5).

[1] For an analytic derivation of the efficient frontier, see (Merton, 1972).

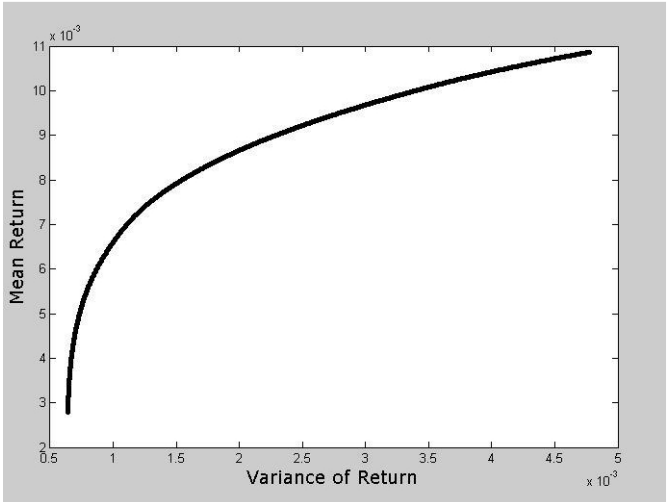


Figure 2.1: The unconstrained efficient frontier of 31-asset universe (Lwin and Qu, 2013).

Obtaining the efficient frontier would simplify the choice of investment for investors and the individual portfolios will be selected based on the investor's risk tolerance and his/her expectation of profit in return. Well spread distribution of portfolios along the efficient frontier provides more alternative suitable choices for investors with different risk-return profiles.

2.2.4 Limitations of the Mean-Variance Model

As with any model, it is crucial to understand the limitations of mean variance analysis in order to use it effectively. Firstly, the mean variance framework was developed for portfolio construction in a *single period*. In the single period portfolio optimization problem, the investor is assumed to make allocations once and for all at the beginning of an investment period, based on the risk and return estimations and correlations of a universe of $\mathcal{N}$ investable assets. Once made, the decisions are not expected to change until the end of the investment period and the impact of decisions arising in subsequent periods is not considered in this case. Hence, the mean variance model essentially represents a passive buy-and-hold strategy (Fabozzi and Markowitz, 2011).

Moreover, the mean variance analysis depends on the perfect knowledge of the expected returns, standard deviation and pair-wise correlation coefficients of all assets under consideration. Chopra and Ziemba (1993) shows that the composition of the optimal portfolio in the mean variance model can be very sensitive to estimation errors in problem inputs. In real world, however, real markets exhibit complexities with unknown and unobservable distributions of returns. Perfect estimates of these inputs are extremely hard, if not impossible, to obtain. Estimating these unknown parameters with free of estimation errors is a whole subject in itself and the mean variance analysis does not address this issue explicitly. Instead, the mean variance model assumes that input parameters provide a satisfactory description of the asset returns. In particular, the first two moments of the distribution (i.e., mean and variance) are considered to be sufficient to correctly represent the distribution of the asset returns and the characteristics of the different portfolios (Crama and Schyns, 2003).

Although Markowitz's mean variance model plays a prominent role in financial theory, direct applications of this model are not of much practical uses for various reasons. It implicitly assumes that the return of assets follows a Gaussian distribution (normal distribution) and investors act in a rational or risk-averse manner. A risk-averse investor prefers the investment with a lower overall risk over the one with a higher overall risk when given two different investments with the same expected return (but different risks). Finally, the model is simplified to be solvable under unrealistic assumptions. Thus, the basic Markowitz model does not reflect the restrictions (constraints) faced by real-world investors (Maringer, 2005). It assumes a perfect market[2] without taxes or transaction costs where short sales are not allowed, and securities are infinitely divisible, i.e. they can be traded in any (non-negative) fraction. It is also assumed that investors do not care about different asset types in their portfolios (Vince, 2007, Chapter 7). These limitations have consequently motivated further developments to improve its applicability in real-world (see Section 2.3.1).

[2] A market is considered to be *perfect* if and only if every possible combination of allocation of assets in a portfolio is attainable.

2.3 An Alternative to Mean-Variance Model

The mean variance analysis reflects risk as the variance or standard deviation of a portfolio. Variance is a statistical measure of the dispersion of returns around the arithmetic mean or average return (the average of squared deviations from the mean). Risk in this context can be described as an indicator of how frequently and by how much the true portfolio return is likely to deviate from its mean. This measure of risk is not practical because the risk of obtaining a result that is above average is considered in the same way as the risk of obtaining a result that is below average. In reality, rational investors' perception against risk is skewed (not symmetric around the mean) as they are more concerned with under-performance rather than over-performance in a portfolio. Variance as a risk measure has thus been widely criticized by practitioners due to its symmetrical measure by equally weighting desirable positive returns against undesirable negative ones (Grootveld and Hallerbach, 1999). This gives rise to research directions where realistic risk measures are used to separate undesirable downside movements from desirable upside movements (Biglova et al., 2004). Among those alternative risk measures which account for the asymmetric nature of risk, Value-at-Risk (VaR) (Morgan, 1996) is a popular risk measure adopted by financial institutions.

2.3.1 Value-at-Risk

Value at Risk (VaR) measures the maximum likely loss of a portfolio from market risk with a given confidence level $(1 - \alpha)$ over a certain time interval. For instance, if a daily VaR is valued as 100,000 with 95% confidence level, this means that during the next trading day there is only a 5% chance that the loss will be greater than 100,000. The higher the confidence level, the better chances that the actual loss will be within the VaR measure. Therefore, the confidence level $(1 - \alpha)$ is usually high, typically 95% or 99%. Formally, the VaR at confidence level $(1 - \alpha)$ 100 % is defined as the negative of the lower α-quantile of the return distribution:

$$VaR_\alpha(\text{R}) = \text{- inf } \{\bar{r} \mid Prob(R \leq \bar{r}) \geq \alpha\}$$

where $\alpha \in (0, 1)$, R is a random portfolio return (Kim et al., 2012; Stoyanov et al., 2013).

2.3.2 Multi-objective Mean-VaR Model

Let us assume that each time t denotes a different scenario and let r_{it} be the observed return of asset i at time t using historical data over the time series horizon T. Let w_i be the proportion of the budget invested in asset i. Given a set of $\mathcal{N}$ assets, the portfolio's return under scenario t is estimated by:

$$\kappa_t(w) = \sum_{i=1}^{\mathcal{N}} \bar{r}_{it} w_i, \qquad t = 1, \ldots, T. \tag{2.12}$$

Let ρ_t be the probability of scenario occurrence and assume all scenarios are considered to have equal probability (i.e., $\rho_t = 1/T$). The expected return of the portfolio is obtained by:

$$\mu(w) = \sum_{t=1}^{T} \kappa_t(w)\rho_t \tag{2.13}$$

The VaR at a given confidence level $(1 - \alpha)$ is the maximum expected loss that the portfolio will not be exceeded with a probability α:

$$\psi(w) = VaR_\alpha(w) = -inf \left\{ \kappa_{t_\alpha}(w) \mid \sum_{j=1}^{t_\alpha} \rho_j \geq \alpha \right\} \tag{2.14}$$

where returns $\kappa_t(w)$ are placed in an ascending order such that $\kappa_{(1)}(w) \leq \kappa_{(2)}(w) \leq \ldots \leq \kappa_{(T)}(w)$ (Anagnostopoulos and Mamanis, 2011a). The negative sign is used in Eq. (2.14) to denote the expected loss since $\kappa_t(w)$ represents the expected return.

The mean-VaR portfolio selection problem is summarized as follows:

$$\begin{aligned} & min \quad \psi(w) \\ & max \quad \mu(w) \\ s.t \quad & \sum_{i=1}^{\mathcal{N}} w_i = 1, \quad 0 \leq w_i \leq 1 \end{aligned} \tag{2.15}$$

2.4 Real-world Constraints

The standard mean variance model is based on several simplifying assumptions. The basic model assumes a perfect market where securities are traded in any (non-negative) fractions, there is no limitation on the number of assets in the portfolio, investors have no preference over assets and they do not care about different asset types in their portfolios. In practical investment management, however, a portfolio manager often faces a number of constraints on his/her investment portfolio for various reasons, such as legal restrictions, institutional features, industrial regulations, client-initiated strategies and other practical matters (Skolpadungket et al., 2007). For example, a portfolio manager may face restrictions on the maximum capital allocation to a particular industry or sector. As a result, the basic model can be extended with a number of real-world constraints to better reflect practical applications. In this section, we describe constraints that are often used in practical applications.

2.4.1 Cardinality Constraint

In the standard model, proportions of assets are not limited no matter how small allocation of the investment is. Very often in practice, investors prefer to have a limited number of assets included in their portfolio since the management of many assets in the portfolio is tedious and hard to monitor. They also intend to reduce transaction costs and/or to assure a certain degree of diversification by limiting the number of assets ($\mathcal{K}$) in their portfolios (Skolpadungket et al., 2007).

Cardinality constraint limits the number of assets that compose the portfolio:

$$\sum_{i=1}^{\mathcal{N}} s_i = \mathcal{K}, \tag{2.16}$$

where binary decision variables $s_i (i = 1, \ldots, \mathcal{N})$ are introduced to indicate if asset i is included in the portfolio. $\mathcal{K}$ is a positive integer less than the number of assets in the investment universe ($\mathcal{N}$).

In the literature, there are two variants of cardinality constraint. One variant is the equality constraint as noted in Eq. (2.16) where cardinality constraint imposes the number of securities in the portfolio to be exactly $\mathcal{K}$ (Armananzas and Lozano, 2005; Chang et al., 2000, 2009; Cura, 2009; Deng et al., 2012; Fernández and Gómez, 2007; Golmakani and Fazel, 2011; Jobst et al., 2001; Skolpadungket et al., 2007; Soleimani et al., 2009; Woodside-Oriakhi et al., 2011). Another variant is inequality constraint (i.e., $\sum_{i=1}^{\mathcal{N}} s_i \leq \mathcal{K}$ or $\mathcal{K}_L \leq \sum_{i=1}^{\mathcal{N}} s_i \leq \mathcal{K}_U$) where cardinality constraint is relaxed with lower and/or upper bounds $[\mathcal{K}_L, \mathcal{K}_U]$ (Anagnostopoulos and Mamanis, 2011b; Cesarone et al., 2013; Chiam et al., 2008; Crama and Schyns, 2003; Gaspero et al., 2011; John, 2014; Liagkouras and Metaxiotis, 2014; Maringer and Kellerer, 2003; Schaerf, 2002). Alternatively, cardinality constraint can be addressed as one of the minimization objectives in the portfolio optimization problem. Anagnostopoulos and Mamanis (2010) consider the portfolio optimization problem as a tri-objective optimization problem in order to achieve the trade-offs between risk, return and the number of securities in the portfolio.

2.4.2 Floor and Ceiling Constraints

The floor and ceiling constraints specify the minimum and maximum limits on the proportion of each asset that can be held in a portfolio (Chang et al., 2000). The former prevents excessive administrative costs for very small holdings, which have negligible influence on the performance of the portfolio, while the latter rules out excessive exposure to a specific asset and, in some cases, it

is restricted by institutional policies. The floor and ceiling constraints are also known as bounding or quantity constraints. Using finite lower and upper bounds, ϵ_i and δ_i respectively, and the binary variable s_i, the floor and ceiling constraints can be represented as follows:

$$s_i = \begin{cases} 1 & \textbf{if the } i^{th} \ (i = 1, \ldots, \mathcal{N}) \text{ asset is held} \\ 0 & \textbf{otherwise}, \end{cases} \tag{2.17}$$

$$\epsilon_i s_i \ \leq \ w_i \ \leq \ \delta_i s_i, \quad i = 1, \ldots, \mathcal{N}, \tag{2.18}$$

Since budget constraint of the basic model requires all weights to sum up to one (see Eq. (2.3)), the sum of lower bounds should not be above one, $\sum_{i=1}^{\mathcal{N}} \epsilon_i \geq 1$, and the sum of upper bound should not be below 1, $\sum_{i=1}^{\mathcal{N}} \delta_i \leq 1$. Since short sales are not allowed in the basic model, floor constraints override Eq. (2.4).

2.4.3 Round Lot Constraint

Many real-world applications require that securities are traded as multiples of minimum lots or batches. Round lot constraint requires the number of any asset in the portfolio to be in an exact multiple of the normal trading lots (Golmakani and Fazel, 2011; Lin and Liu, 2008; Skolpadungket et al., 2007; Soleimani et al., 2009; Streichert et al., 2004a,b). It overcomes the assumption of infinite divisibility of assets the basic model (Jobst et al., 2001). If y_i represents the positive integer variables and ϑ_i is the minimum tradable lot that can be purchased for each asset, the round lot constraint can be stated as follows:

$$w_i = y_i \, . \, \vartheta_i, \quad i = 1, \ldots, \mathcal{N}, \quad y_i \in \mathbb{Z}_+ \tag{2.19}$$

In the literature, round lot constraints are mainly modelled in two variants (see Di Tollo and Roli (2008); Mansini et al. (2014) for detailed classification). In

this work, round lot constraint is modelled as a fraction ϑ_i of the total invested portfolio wealth. In other words, the round lot constraint defined in Eq. (2.19) imposes that each weight must be the multiple of a given fraction ϑ_i where lot size ϑ_i is uniform for all assets. This approach is also adopted by Jobst et al. (2001) and Streichert et al. (2004a,b,c).

The inclusion of round-lot constraint may require relaxation of the budget constraint as the total capital might not be the exact multiples of the minimum trading lot prices for various assets.

2.4.4 Pre-assignment Constraint

The pre-assignment constraint is usually used to model the investor's subjective preferences. An investor may intuitively wish a specific set of assets (Z) to be included in the portfolio, with its proportion to be determined (Chang et al., 2000; Di Tollo and Roli, 2008). This constraint can be modelled with binary variables z_i such that assets that need to be pre-assigned in a portfolio are denoted with one (Gaspero et al., 2011).

$$z_i = \begin{cases} 1 & \textbf{if} \quad i \in Z \\ 0 & \textbf{otherwise}, \end{cases} \tag{2.20}$$

$$s_i \geq z_i, \quad i = 1, \ldots, \mathcal{N}, \tag{2.21}$$

2.4.5 Class Constraints

In practice, investors may ideally want to partition the available assets into mutually exclusive sets (classes). Each set may be grouped with common features or types such as health care assets, energy assets, etc. or grouped by investors' own intuition. Investors may prefer to select at least one asset from each class to construct a well-diversified and/or safe portfolio. Let $C_m, m = 1, \ldots, \mathcal{M}$, be $\mathcal{M}$ sets of asset classes that are mutually exclusive, i.e., $C_i \cap C_j = \emptyset, \forall i \neq j$. Class constraint requires that at least one asset from each class are invested in a

portfolio and can be defined as follows:

$$s_i \in C_m, \quad m = 1, \ldots, \mathcal{M}, \tag{2.22}$$

2.4.6 Class Limit Constraints

Investors may also want to restrict on how concentrated the investment portfolio can be in a particular class or sector. Similar to the floor and ceiling constraints, class limit constraints require that the total proportion invested in each class lies between lower and upper limits specified by the investors. Let L_m be the lower bound and U_m be the upper bound for class m then the class limit constraints are formulated as follows:

$$L_m \leq \sum_{s_i \in C_m} w_i \leq U_m, \quad m = 1, \ldots, \mathcal{M}, \tag{2.23}$$

Note that class constraints (see Section 2.4.5) can be implicitly defined by class limit constraints when a lower bound of each class is defined to be positive. In this case, at least one asset from each class is required to be included in a portfolio. Class and class limit constraints are first introduced by Chang et al. (2000) and Anagnostopoulos and Mamanis (2011a) and Vijayalakshmi Pai and Michel (2009) consequently consider the class constraints in their work. In their studies, class constraints are implied by assuming that $L_m > 0$ for every class $m(m = 1, \ldots, \mathcal{M})$, .

2.4.7 Transaction Costs

When an investor buys or sells securities, expenses are incurred due to brokerage costs and taxes. In general, these costs could be variable and/or proportional to the traded volume. In some cases, a variable fee proportional to the traded amount (Akian et al., 1996; Davis and Norman, 1990; Dumas and Luciano, 1991; Shreve and Soner, 1994) might be imposed and/or they may also come together with a fixed cost (i.e. fixed fee per transaction) (Lobo et al., 2007; Oksendal and Sulem, 2002). Maringer (2005) presents four variants of transaction costs:

fixed only, proportional only, proportional with lower bound and proportional plus fixed costs. Let $y_i \in \mathbb{N}_0^+$ be the natural, non-negative number of asset $i \in [1, \ldots, \mathcal{N}]$ and η_i be its current price When an investor faces proportional costs of ζ_p and/or fixed minimum costs of ζ_f, the transaction cost TC_i of asset i can be expressed as such:

$$TC_i = \begin{cases} \zeta_f & \text{, fixed cost only} \\ \zeta_p.y_i.\eta_i & \text{, proportional cost only} \\ max\{\ \zeta_f,\ \zeta_p.y_i.\eta_i\ \} & \text{, proportional cost with lower limit} \\ \zeta_f + \zeta_p.y_i.\eta_i & \text{, proportional plus fixed cost} \end{cases} \tag{2.24}$$

2.4.8 Turnover and Trading Constraints

This thesis is mainly concerned with the single-period portfolio selection problems. For the sake of completeness, we present variants of constraints that occur in the multi-period formulation of portfolio selection problems. Crama and Schyns (2003) introduces these constraints as a variant of the single-period formulation. Turnover constraints define maximum trading limits pre-specified by practitioners to safeguard against excessive transaction costs between trading periods (Scherer and Martin, 2005)and can be described as follows (Crama and Schyns, 2003):

$$max(w_i - w_i^{(0)}, 0) \leq \overline{B}_i, \quad i = 1, \ldots, \mathcal{N} \tag{2.25}$$

$$max(w_i^{(0)} - w_i, 0) \leq \overline{S}_i, \quad i = 1, \ldots, \mathcal{N} \tag{2.26}$$

where $w_i^{(0)}$ denotes existing proportion of asset i prior to the portfolio construction, $\overline{B}_i$ denotes the maximum purchase and $\overline{S}_i$ denotes maximum sale of asset i.

Trading constraints impose minimum limits to prevent buying and selling tiny

quantities of assets when there are high fixed transaction costs. Trading constraints can be expressed as follows (Crama and Schyns, 2003):

$$w_i = w_i^{(0)} \ \vee \ w_i \geq w_i^{(0)} + \underline{B_i}, \quad i = 1, \ldots, \mathcal{N} \tag{2.27}$$

$$w_i = w_i^{(0)} \ \vee \ w_i \leq w_i^{(0)} - \underline{S_i}, \quad i = 1, \ldots, \mathcal{N} \tag{2.28}$$

where $w_i^{(0)}$ represents existing proportion of asset i in the initial portfolio, $\underline{B_i}$ and $\underline{S_i}$ denote the minimum purchase and sale of asset i respectively.

2.5 Datasets

Problem instances for Mean-Variance model

Test problems based on well-known major market indices for the portfolio optimization problems are publicly available from the OR-library (Beasley, 1990, 1999). Table 2.1 shows the details of these benchmark indices and their sizes. It should be noted that, for commercial reasons, these datasets have been disguised, such that the identities of the assets associated to the data are not unfold. In the current literature of portfolio optimization problems, these market indices provided by the OR-library have been widely used, and are recognized as the benchmark to evaluate the performance of different computational algorithms.

Instance	Origin	Name	Number of assets
D1	Hong Kong	Hang Seng	31
D2	Germany	DAX100	85
D3	UK	FTSE 100	89
D4	US	S&P 100	98
D5	Japan	Nikkei	225
D6	US	S&P 500	457
D7	US	Russell 2000	1318

Table 2.1: The benchmark instances from OR-library.

The first five datasets (D1 – D5) built from weekly price data from March 1992 to September 1997 and their best known optimal solutions are available at: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html. They were first introduced by Chang et al. (2000). The remaining two datasets were built based on the index tracking problem and they were first introduced by Canakgoz and Beasley (2009). These two datasets (D6 and D7) are available at: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/indtrackinfo.html. An example OR-library dataset is also provided in Appendix B.1.

The first five datasets (D1 – D5) have been used for the mean variance constrained portfolio optimization problems considered in chapter 4 and chapter 5. All seven datasets (D1– D7) have been used for mean variance constrained portfolio optimization problems considered in chapter 6.

It should also be noted that Cesarone et al. (2011, 2013) also provide five additional market indices: EuroStoxx50 in Europe, FTSE 100 in UK, MIBTEL in Italy, S & P 500 in USA and NASDAQ in USA. These instances built from weekly price data from March 2003 to March 2008 are publicly accessible at: http://w3.uniroma1.it/Tardella/datasets.html. However, these problem instances are not very well-known and they have not been widely used by many studies.

Problem instances for mean-VaR model

In this research, two new datasets (DS1 and DS2) were created for the mean-VaR portfolio optimization problems studied in chapter 7. These two datasets based on historical daily financial market data have been retrieved from the Yahoo! Finance[3]. It was observed that historical time series downloaded from this site had some missing data points and hence those assets with missing data points were discarded. The first dataset (DS1) consists of 94 securities from the S & P 100 and covers daily financial time series data over a period of three years from

[3] http://finance.yahoo.com

01/03/2005 to 20/02/2008, totalling 750 trading days.

The second dataset (DS2) is composed of 475 securities from the S & P 500 and covers daily financial time series data over a period of one year from 11/04/2013 to 04/04/2014, totalling 250 trading days. The datasets are available to access online at: http://www.cs.nott.ac.uk/~ktl. An example of a small set of dataset is also presented in Appendix B.2. Constituents of datasets DS1 and DS2 are provided in Table B.4 and Table B.5 respectively. These datasets have been used for mean-VaR portfolio optimization with cardinality, quantity, preassignment, round lot, class and class limit constraints in order to study the performance of the evolutionary algorithms considered in this work presented in chapter 7.

2.6 Summary

In this chapter, we provide a detailed description of the various optimization approaches for the mean variance portfolio optimization problems. In addition, the basic concepts and limitations of the mean variance (MV) model are also discussed. An alternative risk measure, value-at-risk (VaR), for the Mean-VaR model is also described. Additionally, practical trading constraints commonly faced by investors are described. The detailed descriptions of the market indices used in this thesis for computational analysis are also presented. This chapter provides an introduction to the background of the constrained portfolio optimization problems considered in this thesis.

Chapter 3

Evolutionary Algorithms: An Overview

> "... one general law, leading to the advancement of all organic beings, namely, multiply, vary, let the strongest live and the weakest die."
>
> Charles Darwin

3.1 Introduction

An optimization problem can be roughly defined as hard if it cannot be solved to optimality, or to any guaranteed bound, by any exact (deterministic) method within a *"reasonable"* computational time (Boussaïd et al., 2013). In the domains of Artificial Intelligence and Operation Research, a metaheuristic, first introduced by Glover (1986), refers to an algorithm designed to approximately solve a wide range of hard optimization problems with little or no modification (Blum et al., 2011; Blum and Roli, 2003; Boussaïd et al., 2013). The term *"meta"* is prefixed to denote that these algorithms are higher-level heuristics, in contrast to problem-specific heuristics (Boussaïd et al., 2013; Talbi, 2009). In the domains of computer science and optimization, a heuristic refers to the art of

discovering new techniques which, especially in practice, deliver good solutions to a problem based on a *"rule of thumb"* or a set of rules derived from domain knowledge (Blum et al., 2011).

Metaheuristics are one of the successful alternative approaches to solve hard optimization problems for which no deterministic methods are known (Boussaïd et al., 2013). However, they are not function optimizers. That is, their goal is to find good solutions to the problem, rather than a guaranteed optimal solution. Metaheuristic algorithms are mainly divided into trajectory-based and population-based algorithms. The former relies on a single solution while the latter manages a set of solutions (population) to perform the search.

Evolutionary Algorithms (EAs) are one of the most studied population-based methods. They are inspired from the process of natural evolutionary principles (Darwin, 1859) in order to develop search and optimization techniques for solving complex problems. Because of their abilities to tackle complex and real-world optimization problems in many different application areas, EAs have gained significant amount of research interest over the last few decades. Multi-objective Evolutionary Algorithms (MOEAs) are one of the current trends in developing EAs.

This chapter firstly introduces some main concepts and definitions related to multi-objective optimization problems. The principles of a number of well-known and commonly used evolutionary algorithms are then presented. It is noted that the scope of this thesis is limited to population-based EAs.

3.2 Multi-objective Optimization Problems

Optimization refers to finding the best possible solution to a problem given a set of limitations or constraints (Coello and Zacatenco, 2006). Multi-objective optimization problems (MOPs) involve multiple performance criteria or objectives which need to be optimized simultaneously (Fonseca and Fleming, 1995).

A general multi-objective optimization problem (MOP) can be formally defined as follows:

$$\left.\begin{aligned} Maximize/Minimize \quad & F(X) = [f_1(X), f_2(X), \ldots, f_J(X)] \\ subject\ to \quad & b_\imath(X) \geq 0, \quad \imath = 1, 2, \ldots, I, \\ & h_e(X) = 0, \quad e = 1, 2, \ldots, E, \\ & X \in \Omega, \quad J \geq 2, \end{aligned}\right\} \tag{3.1}$$

where Ω is a decision space and X is a vector of $\mathcal{D}$ decision variables: $X = [x_1, x_2, \ldots, x_\mathcal{D}]$; J is the number of objectives; I is the number of inequality constraints; and E is the number of equality constraints. The vector of decision variables X can be either continuous or *discrete*. If X is a discrete (and finite) set of solutions, then the problem defined in Eq. (3.1) is called a multi-objective combinatorial optimization problem. $F(X)$ consists of J objective functions $f_j : \Omega \rightarrow \Re$, a mapping from decision variables $[x_1, x_2, \ldots, x_\mathcal{D}]$ to objective vectors $[y = a_1, a_2, \ldots, a_J]$, where $\Re^J$ is the objective space (Coello et al., 2007; Deb, 2001; Zhou et al., 2011).

There are J objective functions considered in Eq. (3.1) and each objective function can be either *minimized* or *maximized*. In the context of optimization, the *duality principle* (Deb, 2001, 2012) suggests that a maximization problem can be converted into a minimization one by multiplying the objective function with -1. This principle has made the optimization problems with mixed type of objectives easy to handle by transforming the objective into one same type of optimization problems.

3.2.1 Pareto optimality

In many real-world applications, the objectives of MOPs are usually conflicting and optimizing one objective often results in degrading the others. The optimal solution for MOPs, therefore, is not a single solution but a set of *'compromise'*

solutions representing the trade-offs (i.e., Pareto set) between the conflicting objectives (Deb, 2001; Fonseca and Fleming, 1995). Before we discuss further, let us present the following definitions (Deb, 2001; Zitzler et al., 2010) that are integral concept in solving MOPs.

Definition 3.1. *A solution X that satisfies all of the $(I + E)$ constraints and variable bounds $X \in \Omega$ is called a **feasible solution**.*

Definition 3.2. *A feasible solution X_1 is defined to **dominate** another feasible solution X_2 (denoted as $X_1 \preceq X_2$ (Deb, 2001)), if both of the following conditions hold:*

1. *The solution X_1 is no worse than X_2 in all objectives.*
2. *The solution X_1 is strictly better than X_2 in at least one objectives.*

Alternatively, it can be stated that X_1 is ***non-dominated*** by X_2 or X_2 is ***dominated*** by X_1.

Definition 3.3. *Two solutions, X_1 and X_2, are called **incomparable**(denoted as $X_1 \parallel X_2$) if neither X_1 dominates X_2 or X_2 dominates X_1 (i.e., if $X_1 \npreceq X_2 \vee X_2 \npreceq X_1$).*

Definition 3.4. *A solution $X' \in \Omega$ is called (globally) **Pareto optimal** or **efficient** if there is no solution $X \in \Omega$ such that $F(X)$ dominates $F(X')$.*

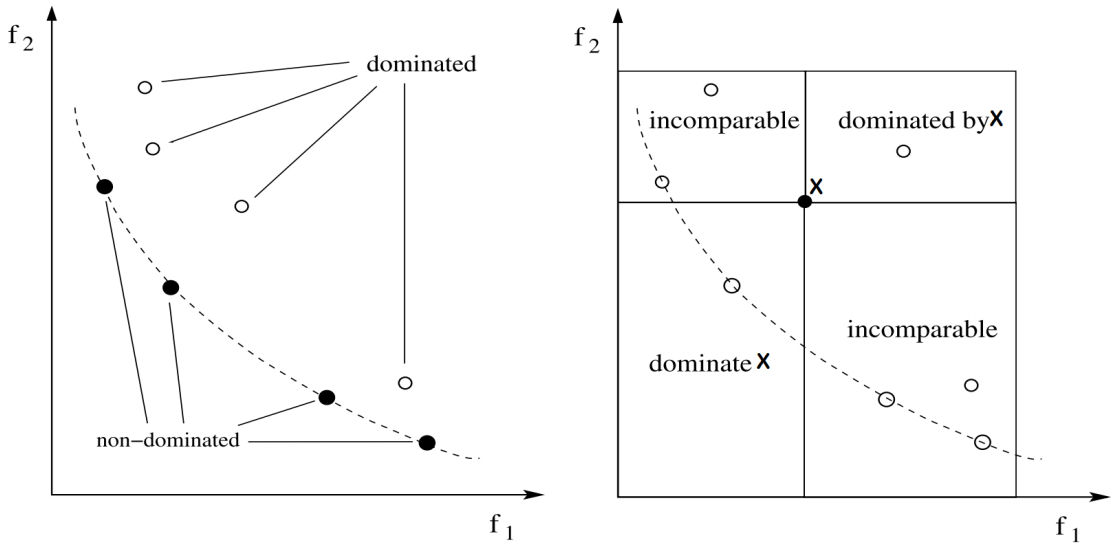
Definition 3.5. *The set of all the Pareto optimal solutions is called the **Pareto set** or **efficient set**, denoted as $\mathbb{P}_{true}$:*

$$\mathbb{P}_{true} = \{X' \in \Omega \mid \nexists X \in \Omega, F(X) \preceq F(X')\}.$$

*The image of the $\mathbb{P}_{true}$ plotted in the objective space is called the **Pareto front** or **efficient frontier**, denoted as EF_{true}:*

$$EF_{true} = \{F(X) \mid X \in \mathbb{P}_{true}\}.$$

Figure 3.1 shows the Pareto optimality concept for a bi-objective minimization problem. Figure 3.1(a) describes the Pareto optimal solutions with filled circles whereas the solutions that are dominated are represented by the non-filled circles. Figure 3.1(b) shows that there exist solutions that are worse than X in both objectives, better than X in both objectives, and incomparable (better in one objective, worse in the other objective).



(a) Non-dominated solutions (b) Dominance relations in reference to **X**

Figure 3.1: Pareto optimality concept for bi-objective minimization problem (Baños et al., 2009).

3.2.2 Multi-objective Optimization Approaches

There are two general approaches to solve the multi-objective optimization problems. One common approach is optimizing all objectives *simultaneously* based on the dominance relationship to determine the Pareto optimal set ($\mathbb{P}_{true}$) or a representative subset of Pareto optimal set (see Section 3.2.1). An alternative approach is to combine the individual objective functions into a single composite function by adopting a weighted sum method as follow.

Weighted Sum Method

Prior to optimization, the weighted sum method transforms the multiple objectives into a single objective function by aggregating all objectives in a weighted function:

$$\left.\begin{aligned} Maximize/Minimize \quad & F(X) = \sum_{j=1}^{J} \lambda_j f_j(X), \\ subject\ to \quad & \lambda_j \geq 0, \\ & \sum_{j=1}^{J} \lambda_j = 1, \end{aligned}\right\} \tag{3.2}$$

where the weights (λ_j) can reflect the relative importance of the objectives. This approach produces a single solution with a given weight vector $\{\lambda_1, \lambda_2, \ldots, \lambda_J\}$. Therefore, the problem must be solved repeatedly with different combination of weights (i.e., pre-determined) in order to achieve multiple solutions to determine the Pareto optimal set ($\mathbb{P}_{true}$) or a representative subset of Pareto optimal set ($\mathbb{P}_{known}$). The main drawback of this approach is that it requires a priori knowledge about the relative importance of the objectives (Konak et al., 2006).

3.2.3 Optimization Goals of MOPs

The ultimate goal of a MOP is to identify the set of Pareto solutions ($\mathbb{P}_{true}$). The Pareto front gives a set of reasonable choice and it is a choice of the decision maker to pick a point along the Pareto front as his/her ultimate solution. However, identifying the entire Pareto set ($\mathbb{P}_{true}$) is practically impossible for large-scale multi-objective optimization problems. In fact, for many MOPs, especially for combinatorial optimization problems, proof of optimal solutions is computationally infeasible. In such cases, a practical approach is to investigate a set of solutions (*the best-known Pareto set*) that best approximate the true Pareto front ($\mathbb{P}_{true}$) (Konak et al., 2006).

3.3 Evolutionary Algorithms

Evolutionary Algorithm (EA) is a collective term for all variants of optimization algorithms that are inspired by biological evolution. An evolutionary algorithm (EA) is an iterative and stochastic (involving random variables) process that operates on a set of individuals (*population*) through operations of *selection*, *recombination* and *mutation*, thereby producing better solutions. A generic structure of an EA is described in Algorithm 3.1 (Bäck and Schwefel, 1993).

Algorithm 3.1: Generic Evolutionary Algorithm

```
1 g ← 0;
2 initialize a population P^g with random individuals;
3 evaluate each individual in P^g;
4 while not termination condition do
5     g ← g + 1;
6     P̄^g ← recombine(P^g);
7     P̂^g ← mutate(P̄^g);
8     evaluate(P̂^g);
9     P^{g+1} ← select(P̂^g ∪ P^g);
```

An *individual* represents a potential solution to the problem being solved. Initially, the population is generated randomly or with the help of problem-specific heuristics. Each individual in the population is evaluated by a fitness function, which is a measure of quality with respect to the problem under consideration. At each iteration (*generation*), a population of candidate solutions is capable of reproducing and is subject to genetic variations followed by the environmental pressure that causes natural selection (survival of the fittest). New offspring solutions are produced by *recombination* of parents and *mutation* of the resulting individuals to promote diversity. A suitable selection strategy is then applied to identify the solutions that survive to the next generation. This process repeats until a predefined number of generations (or function evaluations) or some other specific stopping criteria are met (Boussaïd et al., 2013).

3.3.1 Single Objective Evolutionary Algorithms

This section reviews the principles and applications of a number of population-based evolutionary algorithms for single objective optimization approaches. These EAs may be adapted or hybridized to solve the portfolio optimization problems concerned in this thesis.

3.3.1.1 Population-Based Incremental Learning

Population-based incremental learning (PBIL), a combination of evolutionary algorithm and competitive learning, was first introduced by Baluja (1994). PBIL abstracts away from the crossover and selection operators and achieves its search through probability estimation and sampling techniques. The main feature of PBIL is the introduction of a real-valued probability vector V which is explicitly utilized to generate promising solutions. It maintains the probability vector V characterizing the structures of high-quality solutions found throughout the evolution. The procedure of the standard PBIL is shown in Algorithm 3.2 (Baluja, 1994).

Given a $\mathcal{D}$-dimensional binary optimization problem, PBIL maintains a $\mathcal{D}$-dimensional probability vector $V := \{v_1^g, \ldots, v_{\mathcal{D}}^g\}$. The i^{th} element of V represents the probability that the i^{th} element of a candidate solution will be equal to 1. Initially, the values of the probability vector are initialized to 0.5 to reflect the lack of a priori information of each variable, and sampling from this vector will thus create a uniform distribution of the initial population on the feasible parameter space (Yang et al., 2007). In each generation g, the probability vector v^g is utilized to generate a set S of n candidate solutions. Each solution in set S is then evaluated and assigned a fitness value using a problem-specific fitness function. After the fitness evaluation, the probability vector is updated by shifting towards the best so far solution $B^g = \{b_1^g, \ldots, b_{\mathcal{D}}^g\}$ as follows:

$$v_i^g = (1 - \mathcal{LR}) \times v_i^g \; + \; \mathcal{LR} \times b_i^g \; ; \quad i = 1, \ldots, \mathcal{D}, \tag{3.3}$$

Algorithm 3.2: The basic procedure of PBIL

Input: $\mathcal{D}$: the number of dimension in probability vector,
$\mathcal{LR}$: learning rate,
$\mathcal{MP}$: mutation probability,
β: the amount of perturbation on probability vector,
n: the number of solutions in set S;
Output: S^g;

$g := 0$;
// initialize probability vector $V := \{v_1^g, \ldots, v_{\mathcal{D}}^g\}$.
for $i := 1$ *to* $\mathcal{D}$ **do**
  v_i^g := 0.5;
repeat
  $g := g + 1$;
  $S^g \leftarrow generate\ n$ samples by V;
  $evaluate$ samples S^g;
  $B^g \leftarrow select$ the best solution from $(B^{g-1} \cup S^g)$;
  // update V towards best solution B^g
  for $i := 1$ *to* $\mathcal{D}$ **do**
    $v_i^g := (1 - \mathcal{LR}) \times v_i^g + \mathcal{LR} \times b_i^g$;
  // mutate V
  for $i := 1$ *to* $\mathcal{D}$ **do**
    if $rand(0, 1] < \mathcal{MP}$ **then**
      $v_i^g := (1 - \beta)v_i^g + \beta \times randint(0, 1)$;
until V *has converged or termination condition is met*;

where $\mathcal{LR}$ is the learning rate specifying the distance the probability vector V is shifted at each generation. At each iteration, a bit-wise mutation operation may then be adopted to maintain diversity and avoid local optima. During this phase, a small amount of probability perturbation β is performed on a subset of the vector V if a certain mutation probability $\mathcal{MP}$ is met:

$$v_i^g = (1 - \beta) \times v_i^g \ + \ \beta \times randint(0, 1), \tag{3.4}$$

As the search progress, the probability vector V is expected to shift gradually to solutions with the highest fitness values.

In contrast to traditional evolutionary algorithms, PBIL stores a single probability vector instead of a large set of solutions (see Figure 3.2). Since it does not need to maintain the population of the solution, PBIL has lowered memory usage than traditional EAs. Moreover, it is computationally less expensive since it does not require to perform crossover and selection operators (Baluja, 1994; Baluja and Caruana, 1995; Southey and Karray, 1999; Ventresca and Tizhoosh, 2008).

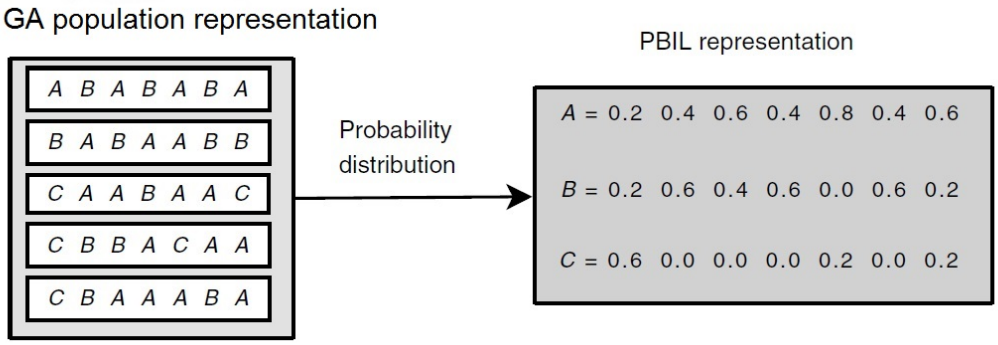


Figure 3.2: Difference between GA and PBIL representation (Gosling et al., 2005; Talbi, 2009).

Due to its straight-forward design philosophy and implementation simplicity, PBILs has gained ample attractions and many studies have been contributed to the literature. Some of these studies are concerned with the applications of the algorithm (Folly, 2011; Galić and Höhfeld, 1996; Gosling et al., 2005; Kern, 2006; Vega-Rodríguez et al., 2007; Xing and Qu, 2011a,b). Others are concerned with the extensions of the method to continuous spaces (Sebag and Ducoulombier, 1998; Yuan and Gallagher, 2003), for multi-objective optimization problems (Bureerat and Sriworamas, 2007), with parallel versions (Baluja, 1997; Yang et al., 2007; Yang and Yao, 2005), and with combination of other methods (Bureerat, 2011; Hong et al., 2008; Quek et al., 2009; Ventresca and Tizhoosh, 2008).

3.3.1.2 Differential Evolution

Differential evolution (DE) algorithm, first introduced by Storn and Price (1997), is one of the most popular evolutionary algorithms for the continuous global optimization problems. DE is a stochastic direct search method that exploits directional information from a population of potential solutions to explore the search space. The main idea of DE is that it uses a scheme (so-called self-referential reproduction scheme) for generating trial population. Selection operation then determines which individuals will survive into the next generation. The detailed procedure of the DE is shown in Algorithm 3.3 (Mezura-Montes et al., 2006).

Algorithm 3.3: Basic DE Algorithm

Input: G: the number of generation,
$\mathcal{D}$: the number of dimension,
$\mathcal{F}$: a scaling factor,
$\mathcal{CR}$: crossover probability,
$\mathcal{NP}$: the number of population;
Output: X_j^{g+1};

$g := 1$;
Generate a random initial population $X_j^g,\ \forall j, j = 1, \ldots, \mathcal{NP}$;
Evaluate $f(X_j^g),\ \forall j, j = 1, \ldots, \mathcal{NP}$;
for $g := 1$ *to* G **do**
  for $j := 1$ *to* $\mathcal{NP}$ **do**
    Select randomly $r_1, r_2, r_3 \in [1, \mathcal{NP}] \wedge\ r_1 \neq r_2 \neq r_3 \neq j$;
    $\mathbb{k} := randint(1, \mathcal{D})$;
    for $i := 1$ *to* $\mathcal{D}$ **do**
      if $((rand_i[0, 1) < \mathcal{CR})$ *or* $(i == \mathbb{k}))$ **then**
        $u_{j,i}^{g+1} := x_{r_3,i}^g\ +\ \mathcal{F} \times (x_{r_1,i}^g - x_{r_2,i}^g)$;
      else
        $u_{j,i}^{g+1} := x_{j,i}^g$;
    if $f(U_j^{g+1}) \leq f(X_j^g)$ **then** `// consider minimization problems`
      $X_j^{g+1} := U_j^{g+1}$;
    else
      $X_j^{g+1} := X_j^g$;
  $g := g + 1$;

Population Structure

DE is a population-based algorithm that is designed to optimize functions in a $\mathcal{D}$-dimensional continuous domain (Simon, 2013). An individual X with $\mathcal{D}$ decision variables represents a potential solution for the optimization problem:

$$\begin{aligned} X =& \{x_1,\ x_2,\ \ldots,\ x_{\mathcal{D}}\}, \\ \epsilon_i^* \leq x_i \leq \delta_i^*,&\quad i = 1, \ldots, \mathcal{D}. \end{aligned} \tag{3.5}$$

where ϵ_i^* and δ_i^* are the lower and upper boundary constraints of the i^{th} dimension respectively. At each generation g, DE maintains a population of $\mathcal{NP}$ individuals, $X_j^g, j \in [1, \mathcal{NP}]$ and the structure of a population is described as follows:

$$\begin{aligned} P^g = \{X_1^g,\ X_2^g,\ \ldots,\ X_{\mathcal{NP}}^g\}, \\ g = 1, \ldots, G;\ \ \mathcal{NP} \geq 4. \end{aligned} \tag{3.6}$$

Initialization

A population of candidate solutions for the optimization task to be solved is randomly initialized within the given lower and upper bounds:

$$\begin{aligned} P^0 = x_{j,i}^0 = \epsilon_i^* + rand_i[0,1) \times (\delta_i^* - \epsilon_i^*), \\ j = 1, \ldots, \mathcal{NP};\ \ i = 1, \ldots, \mathcal{D}. \end{aligned} \tag{3.7}$$

where $rand_i[0,1)$ denotes a uniformly distributed random real value within the range[0,1).

Reproduction

The self-referential population reproduction scheme of DE is different from the other evolutionary algorithms. From the first generation onward, a candidate or trial population for the subsequent generation, $U^g = u_{j,i}^{g+1}$, is generated through

mutation and crossover operations by randomly sampling and combining the current population, P^g.

Mutation

DE mutation operation utilizes the differences of vector formed from chromosomes in the evolving population in order to determine both the degree and direction of perturbation applied to the mutant individual. Figure 3.3 illustrates how the mutant vector V is obtained in a two dimensional parametric space.

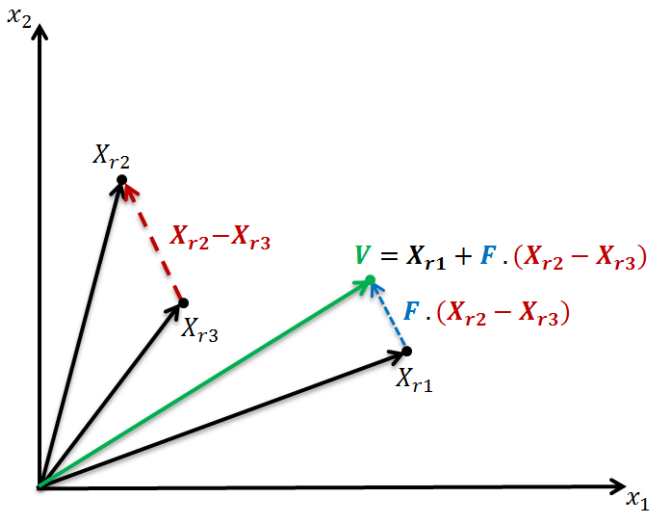
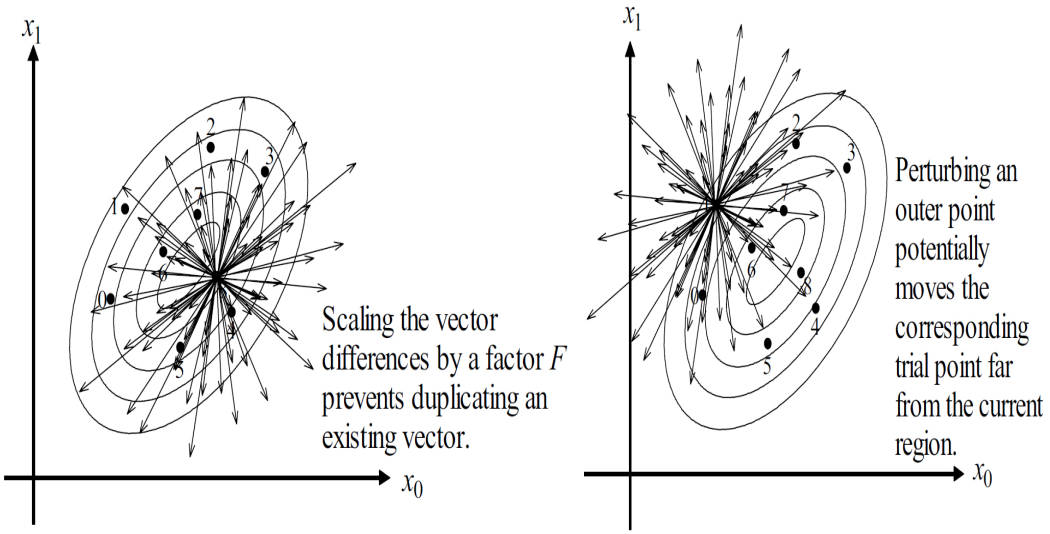


Figure 3.3: Illustration of a basic DE mutation: the weighted differential, $\mathcal{F} \times (X_{r_2} - X_{r_3})$ is added to the based vector, X_{r_1}, to produce a trial vector V (Simon, 2013).

The basic DE mutation scheme adds a scaled vector difference to a third vector as follows:

$$\begin{aligned} V_j^{g+1} &= X_{r_3}^g \; + \; \mathcal{F} \; \times \; (X_{r_1}^g - X_{r_2}^g), \\ & j = 1, \ldots, \mathcal{NP}; \;\; g = 1, \ldots, G; \\ & r_1, \; r_2, \; r_3 \in [1, \mathcal{NP}], \end{aligned} \tag{3.8}$$

where three indexes, r_1, r_2 and r_3, refer to the randomly chosen vectors of the current population. They are mutually exclusive integers randomly chosen from the range [1,$\mathcal{NP}$], which are also different from the index j. A scaling factor, $\mathcal{F} \in (0, 1+)$, is a positive real number that controls the rate at which the population evolves (Price et al., 2006). In addition, scaling can shift the focus of the search between points and can reduce the probability of being trapped in a local minimum. Figure 3.4 shows the features of DE mutation scheme. Figure 3.4(a) illustrates that trial vectors avoid producing duplicate existing points due to the scaling of the vector difference. Figure 3.4(b) illustrates that the probability of being trapped in a local minimum can be reduced due to the presence of substantial number of combination of difference vector.



(a) Effects of scaling (b) Effects of large differences

Figure 3.4: The effects of scaling, and large vector differences (Price et al., 2006).

Crossover/Recombination

DE crossover operation is introduced in order to promote the diversity of the population. The trial vector U_j^g is constructed from components of the parent X_j^g and mutant vectors V_j^g.

$$\begin{aligned} U_j^g = u_{j,i}^{g+1} &= \begin{cases} v_{j,i}^{g+1}, & \textbf{if} \quad rand_i[0,1) \leq \mathcal{CR} \ \vee \ i == \Bbbk \\ x_{j,i}^g, & \textbf{otherwise}, \end{cases} \\ \Bbbk &\in \{1, \ldots, \mathcal{D}\}, \\ j &= 1, \ldots, \mathcal{NP}; \quad i = 1, \ldots, \mathcal{D} \end{aligned} \tag{3.9}$$

where $rand_i[0,1)$ is a uniformly distributed random number generated at each j^{th} index. The index $\Bbbk$ denotes a randomly selected chromosome which is used in order to ensure that each candidate individual, U_j^{g+1}, differs from its counterpart in the previous generation, X_j^g, by at least one parameter. Otherwise, no new individual would be created and the population would not vary (Brest et al., 2006; Lampinen, 2002). $\mathcal{CR} \in [0,1]$ controls the influence of the parent in the generation of the candidate population. Higher $\mathcal{CR}$ value means less influence of the parent.

Selection

During the selection process, each individual from the candidate population is compared with its counterpart in the current population. If the candidate individual is better than or as good as its counterpart in the current population, it wins a place in the population of the next generation, P^{g+1}:

$$X_j^{g+1} = \begin{cases} U_j^{g+1}, & \textbf{if} \quad f(U_j^{g+1}) \ \leq \ f(X_j^g) \\ X_i^g, & \textbf{otherwise}, \end{cases} \tag{3.10}$$

The above greedy selection approach ensures that the population of next generation is at least as good as their counterparts in the current generation. Note

that the candidate individual is compared to only one individual, *not* to all the individuals in the current population.

Constraint Handling

It is important to note that the resultant trial individual may violate the bound constraints as a result of the recombination scheme (Lampinen, 2002; Onwubolu and Davendra, 2006). In such case, the simple way is to replace those indexes that violated the boundary constraints with random values generated within the feasible range as follows:

$$u_{j,i}^{g+1} = \begin{cases} \epsilon_i^* \ + \ rand_i(0,1] \times (\delta_i^* - \epsilon_i^*), & \textbf{if} \quad u_{j,i}^{g+1} < \epsilon_i^* \ \vee \ u_{j,i}^{g+1} > \delta_i^* \\ u_{j,i}^{g}, & \textbf{otherwise}, \end{cases} \tag{3.11}$$

Discrete Optimization

As mentioned in the introduction of this section, DE was originally designed for optimization problems with continuous domain, but it can be modified for discrete domains. Fundamentally there are two approaches to extend DE for discrete problems. We can generate the mutant vector V with the standard DE methods (see Eq. 3.8), and then modify it to lie in the problem domain. Alternatively, we can modify the mutation method in such a way that the generated mutant vector V fall within the problem domain (Simon, 2013).

Variants of DE

There are several variants of DE and it is commonly classified using the *DE/x/y/z* notation where x represents a string denoting the base vector to be perturbed, y indicates the number of difference vectors considered for perturbation of x and z denotes the type of crossover being used. The most popular scheme is called "DE/rand/1/bin", where "rand" refers that the individuals selected to compute

the mutation are chosen at random, "1" represents the number of pairs of solutions chosen and finally "bin" denotes a binomial crossover. Algorithm 3.3 shows the basic "DE/rand/1/bin" scheme.

Applications and Improvements

Since its introduction in 1995, DE has gained significant interests from many researchers and practitioners due to its simplicity and efficiency. This has resulted in many variants of DE algorithm. Some of these variants are devised to tackle specific problems (Cai et al., 2008; Das et al., 2008; Krink et al., 2009; Krink and Paterlini, 2011; Onwubolu and Davendra, 2006; Vasile et al., 2011; Zhang et al., 2013). Others are concerned with self-adapting DE control parameters for numerical optimization (Brest et al., 2006; Ghosh et al., 2011; Qin et al., 2009), with opposition-based DE (Rahnamayan et al., 2008) , with parallel DE (Tasoulis et al., 2004), for constrained optimization (Becerra and Coello, 2006; Mezura-Montes et al., 2010; Mohamed and Sabry, 2012; Zhang et al., 2008), for multi-objective optimization problems (Lampinen, 2002; Robič and Filipič, 2005; Wang and Cai, 2012; Zhang and Sanderson, 2009), using combination of other search algorithms (Fan and Lampinen, 2003; Jia et al., 2011; Liu et al., 2010; Mininno et al., 2011; Noman and Iba, 2008; Yang et al., 2008). Detailed survey of DE and its recent advances can be found in (Das and Suganthan, 2011; Neri and Tirronen, 2010).

3.3.1.3 Scatter Search

Scatter search (SS), first proposed by Glover (1977, 1986), is an evolutionary algorithm that has been successfully applied to a diverse array of hard optimization problems (Glover et al., 2000a). Scatter search algorithm is mainly designed to operate on a small set of solutions (*reference set*) and new solutions are constructed in a *systematic* way by combining subsets of solutions from the reference set ($RefSet$). It uses strategies for search diversification and intensification that have proved effective in a variety of optimization problems. Algorithm 3.4 shows the basic procedure of scatter search (Martí et al., 2006).

Algorithm 3.4: Basic Scatter Search Procedure

Input: $\mathcal{NP}$: the number of population,
b: the number of solutions in $RefSet$;
Output: Best found solution or set of solutions;

1 $P = \emptyset$;
2 Use **diversification generation method** to construct a solution x and apply the *improvement method*. If the resulting solution $x \notin P$ then add x to P. Otherwise, discard x. Repeat this step until $|P| = \mathcal{NP}$;
3 Use the **reference set update method** to build the reference set $RefSet$ with b *best* solutions in P;
4 $NewSolutions$ = TRUE;
5 **while** *(* $NewSolutions$ *)* **do**
6 Generate $NewSubsets$ with **subset generation method**;
7 $NewSolutions$ = FALSE;
8 **while** *(* $NewSubsets \neq \emptyset$ *)* **do**
9 Select the next subset $\bar{s}$ in $NewSubsets$;
10 Apply the **solution combination method** to $\bar{s}$ to obtain one or more new trial solutions;
11 Apply the **improvement method** to the trial solutions;
12 Apply the **reference set update method**;
13 **if** *(* $RefSet$ *has changed)* **then**
14 $NewSolutions$ = TRUE;
15 Delete $\bar{s}$ from $NewSubsets$;

Scatter search algorithm is based on a very flexible framework since each of its components can be implemented in a variety of ways and degrees of sophistication (Martí et al., 2006). The template of a scatter search algorithm generally has five components (Glover, 1998). Figure 3.5 shows the schematic representation of the interaction among these five components.

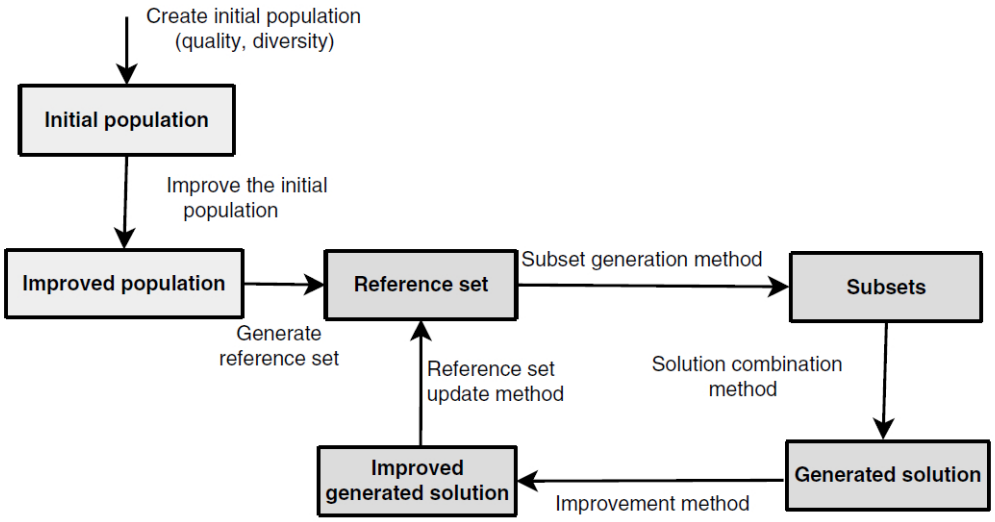


Figure 3.5: Search components of the scatter search algorithm (Talbi, 2009).

Diversification Generation Method:

This method generates a large set P of diverse trial solutions, using an arbitrary trial solution (or seed solution) as an input.

Improvement Method:

This method transforms a trial solution into one or more improved trial solutions. The generated solutions may or may not be better (in terms of quality and feasibility) than the input solution. The diversity of solutions provided by the combination method is exploited by the improvement method in order to achieve enhanced solutions. The possible improvement method ranges from simple variants of local search to a very specialized search (Herrera et al., 2006).

Reference Set Update Method:

This method builds and maintains a reference set with b solutions where the value of b is typically small. The objective is to ensure diversity while keeping high-quality solutions. Several other alternative criteria may be

used to add solutions to the reference set and delete solutions from the reference set (Resende et al., 2010).

Subset Generation Method:
This method operates on the reference set to produce subsets of reference solutions (Martí et al., 2006).

Solution Combination Method:
This method uses subsets of solutions generated from the subset generation method to construct one or more combined solutions. The resulting new solutions are combined from usually two (or more) reference solutions. The combination of reference solutions is specifically designed to exploit context information, not contained separately in the reference solutions. This method also has a strong influence on the exploration since generated new solutions contribute to population diversity (Herrera et al., 2006).

In contrast to other evolutionary algorithms, such as genetic algorithms (GA), which typically have some stochastic element to their solution generation operations, scatter search is designed to minimize (if not eliminate) decisions made by random (or more usually pseudo-random) chance (Burke et al., 2010). Instead, scatter search is based on "*systematic and strategically designed rules*" (Burke et al., 2010; Glover et al., 2000a,b). That is, the solution generation in scatter search replaces randomized implementation with a deterministic method called *Subset Generation Method* in order to construct better solutions (Herrera et al., 2006). This method is usually designed based on the structure and properties of the problem being solved, as well as on the search history (Resende et al., 2010). Another difference from traditional EAs is that new solutions encountered during the evolving process are added to the population or to the reference set. In many GAs, new solutions are allowed to enter the current population based on the solution quality. In scatter search, however, an arbitrary method for comparing the two solutions is used to reflect the reference set's overall diversity. Therefore, the decision that a new solution becomes a member of reference set is not purely based on its quality (objective function values), but also based on

its diversity relative to members of the reference set (Burke et al., 2010; Resende et al., 2010).

Out of the five components, the improvement method is not strictly required. It can be incorporated if high-quality solutions are necessary. When the optional improvement methods are incorporated into the scatter search procedure, it can be classified as a family of memetic algorithms (MAs) (Moscato et al., 2004). Since its introduction, scatter search has been successfully applied to a wide range of applications (Burke et al., 2010; Pinol and Beasley, 2006; Russell and Chiang, 2006). It has also been extended for multi-objective optimization (Beausoleil, 2006; Nebro et al., 2008), with parallel version (Garcıa López et al., 2006) and with other methods (Maenhout and Vanhoucke, 2010).

3.3.2 Pareto-based MOEAs

The idea of measuring an individual's fitness on the basis of Pareto dominance was first proposed by Goldberg (1989). Many approaches to exploiting partial orderings of the population have subsequently been proposed in the literature. This section reviews a number of MOEAs that explicitly use a measure based on Pareto domination to rank individuals. A number of MOEAs discussed in this section are adapted or hybridized for the constrained portfolio optimization problems considered in this thesis.

3.3.2.1 Elitist Non-dominated Sorting Genetic Algorithm

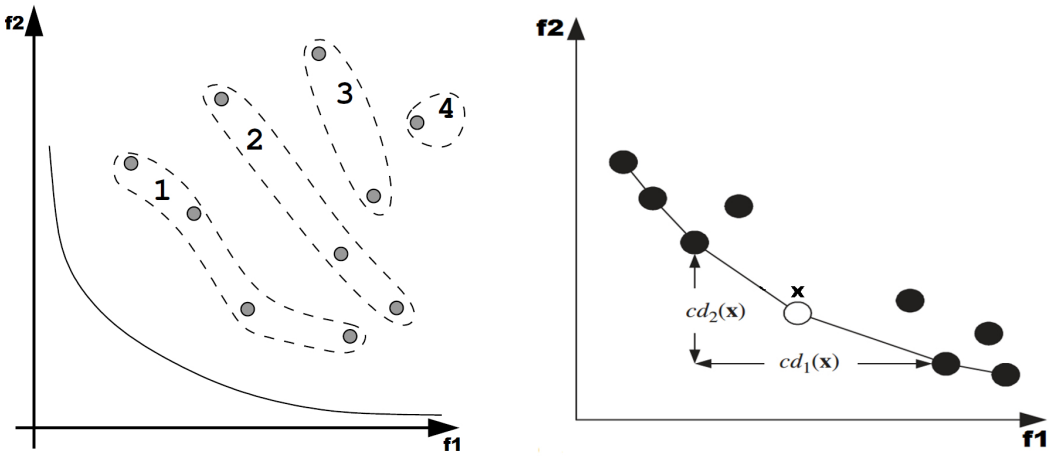
Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II), an improved version of NSGA (Srinivas and Deb, 1994), was presented by Deb et al. (2002). Compared to its predecessor, NSGA-II is enhanced with three significant features:

- A fast non-dominated sorting approach that reduces the computational complexity from O(JI^3) to O(JI^2), where J is the number of objective and I is the population size.

- An elitism approach that prevents the loss of promising solutions already found is introduced.
- A parameterless diversity preservation scheme is introduced by adopting a *crowded-comparison* approach that eliminates the difficulty of setting an appropriate parameter value for the fitness sharing function to ensure the diversity in the population.

Non-dominated Sorting

NSGA-II is built on the basic framework that utilizes a layered classification technique. The basic idea is to classify individuals in the population into non-dominated fronts Ł$_i$. First, all non-dominated individuals in population P are identified. The set of these identified individuals are classified into one category with the first front or level Ł$_1$ and they are then eliminated from further consideration. The process is then repeated with the remaining individuals until the entire population is ranked (see Figure 3.6(a)).



(a) ranking (b) distance

Figure 3.6: Non-dominated sorting and crowding distance methods used in NSGA-II for two objectives (Deb et al., 2002).

The procedure of a faster non-dominated sorting is outlined in Algorithm 3.5. Each individual p has two entities: (1) a domination count n_p, the number of solutions which dominate p, and (2) a set S_p of individuals which dominate p. The individuals in the first front will have their domination count as zero. For each individual p with $n_p = 0$, reduce the domination count of each solution $p' \in S_p$ by one. When the domination count of a member p' in set S_p becomes zero, it is put in a set P' which keeps the second non-dominated front. This process repeats until all fronts are classified.

Algorithm 3.5: Non-dominated Sorting Procedure (Deb et al., 2002).

Input: A set of solutions (population P);
Output: Sorted population P ranked by non-dominating criteria;

forall the $p \in P$ **do**
  $S_p = \emptyset$;
  $n_p = 0$;
  forall the $p' \in P$ **do**
    if $(p \preceq p')$ **then**
      $S_p = S_p \cup p'$;
    else if $(p' \preceq p)$ **then**
      $n_p = n_p + 1$;
  if $(n_p = 0)$ **then**
    $p_{rank} = 1$;
    $Ł_1 = Ł_1 \cup p$;
$i = 1$;
while $(Ł_i \neq \emptyset)$ **do**
  $P' = \emptyset$;
  forall the $p \in Ł_i$ **do**
    forall the $p' \in S_p$ **do**
      $n_{p'} = n_{p'} - 1$;
      if $(n_{p'} = 0)$ **then**
        $p'_{rank} = i + 1$;
        $P' = P' \cup p'$;
  $i = i + 1$;
  $Ł_i = P'$;

Crowding Distance Computation

The crowding distance of a particular solution i is the average distance of its two neighbouring solutions. Figure 3.6(b) shows the crowding distance of individual x which is calculated as an average distance of the largest cuboid enclosing x without including any other point. The crowding distance is computed by first sorting the population in an ascending order of objective function values. The boundary solutions of each objective function are set with infinite values in order to ensure that they are always selected. All other intermediate solutions are computed by the absolute normalized difference of two adjacent solutions. The overall crowding distance is obtained by adding the individual distance values of each objective (Deb et al., 2002). The procedure is shown in Algorithm 3.6 where $f_j(x)$ denotes the j^{th} objective function value of the individual x in the set X, $cd_j(x)$ denotes the crowding distance of j^{th} objective function of individual x and f_j^{max} and f_j^{min} are the maximum and minimum values of the j^{th} objective function.

Algorithm 3.6: Crowding distance assignment (Deb et al., 2002).

Input: A set of n solutions $X = \{x_1, \ldots, x_n\}$;

Output: A set of solutions with crowding distance values ;

`// for each objective` j

1 **for** $j := 1$ *to* J **do**

2 $\quad sort(X, j)$;

3 $\quad cd_j(x_1) = cd_j(x_n) = \infty$; `// boundary points`

4 $\quad$ **for** $i := 2$ *to* $n - 1$ **do**

5 $\quad\quad cd_j(x_i) = cd_j(x_i) + \dfrac{f_j(x_{i+1}) - f_j(x_{i-1})}{f_j^{max} - f_j^{min}}$;

The distance value provides an estimation of the density of solutions surrounding to a particular solution (Deb et al., 2002). During the selection process, in a case that two solutions, x_1 and x_2, are on the same non-dominated front, the crowding distance measure is used as a tie-breaker to choose the winner be-

tween the two. The one with the higher crowding distance is preferred (Konak et al., 2006). This is also termed as crowded tournament selection operator as described below (Deb, 2001).

Definition 3.6. ***Crowded Tournament Selection Operator:*** *A solution X_1 wins a tournament with another solution X_2 if any of the following conditions hold:*

1. *if solution X_1 has better rank than X_2.*
2. *if both have the same rank but solution X_1 has a better crowding distance than solution X_2.*

Algorithm 3.7: Outline of the NSGA-II Procedure (Deb, 2001).

Input: An initial population P^g;
Output: P^{g+1} ;
Step 1: combine current and candidate populations and create $C^g = P^g \cup \bar{P}^g$ perform a non-dominated sorting to C^g and identify different fronts $Ł_i, i = 1, \ldots, n$ (see Algorithm 3.5).
Step 2: Set new population $P^{g+1} = \emptyset$. Set a counter $i = 1$. Until ($\mid P^{g+1} \mid + \mid Ł_i \mid < \mathcal{NP}$), perform $P^{g+1} = P^{g+1} \cup Ł_i$ and $i = i + 1$.
Step 3: perform the crowding distance sorting procedure and add the most widely spread ($\mathcal{NP} - \mid P^{g+1} \mid$) solutions by using the crowding distance values in the sorted $Ł_i$ to P^{g+1}.
Step 4: create candidate population $\bar{P}^{g+1}$ from P^{g+1} using the crowded tournament selection, crossover and mutation operations.

The procedure of the NSGA-II is described in Algorithm 3.7. At each generation g, a current population P^g and a candidate population $\bar{P}^g$ are merged into one as C^g. The combined population C^g is then sorted and ranked by non-domination level. Elitism is therefore ensured by preventing the loss of good solutions already found in the previous population. The new population P is first filled with solutions in the best front $Ł_1$. If the number of solutions in set $Ł_1$ is less than the population size $\mathcal{NP}$, the remaining members of the next population are selected

from subsequent fronts $Ł_2$, $Ł_3$ and so on. To be able to select an exact number of $\mathcal{NP}$ individuals, crowding distance measure is used to decide the winner among the individuals which resides at the same front (see Algorithm 3.7, line-10). The new population $\bar{P}^{g+1}$ is then generated by the mutation, crossover and selection operations.

3.3.2.2 Improving the Strength Pareto Evolutionary Algorithm

Improving the Strength Pareto Evolutionary Algorithm (SPEA2), an improved version of SPEA (Zitzler and Thiele, 1999), was presented by Zitzler et al. (2001). The procedure of the SPEA2 is outlined in Algorithm 3.8.

Algorithm 3.8: SPEA2 Procedure (Zitzler et al., 2001).

Input: G: the number of generation,
A_{size}: the size of archive A;
Output: A'

1 *generate* a population P;
2 *initialize* an archive $A = \emptyset$;
3 **for** *($g = 1$ to G)* **do**
4 *evaluate* fitness of each individual in P^g and A;
5 *copy* all non-dominated individuals in P^g and A to A';
6 **if** *($\mid A' \mid > \mathcal{A}_{size}$)* **then**
7 *apply* truncation operator to reduce the size of archive A' to $\mathcal{A}_{size}$
8 **else if** *($\mid A' \mid < \mathcal{A}_{size}$)* **then**
9 *fill* archive A' with dominated individuals from population P^g
10 *apply* binary tournament selection with replacement on A' to fill the mating pool;
11 *apply* crossover and mutation operations to the mating pool and set P^{g+1} to the resulting population;
12 $g = g + 1$;

The main improved features of SPEA2 are:

- A fine-grained fitness assignment scheme which records for each individual the number of individuals that it dominates and it is dominated by.
- A nearest neighbour density estimation technique, which promotes a more precise guidance of the search space.
- A new archive truncation method that ensures the preservation of boundary solutions.

Fine-grained Fitness Assignment Strategy

SPEA2 is among the first evolutionary algorithms that introduces elitism by explicitly maintaining an external population. It incorporates a fine-grained fitness assignment strategy. At each generation, all non-dominated individuals are copied to an archive A. For each individual x in the archive A and the population P, the *strength* value $S(x)$ representing the number of solutions it dominates is evaluated as:

$$S(x) = \mid \{y | y \in P \cup A \quad \wedge \quad x \preceq y\} \mid \tag{3.12}$$

where $\mid . \mid$ denotes the cardinality of a set. The raw fitness of an individual x is then evaluated based on the strength values $S(x)$ as:

$$\Psi(x) = \sum_{y \in P \cup A,\ y \preceq x} S(x) \tag{3.13}$$

The fitness value of non-dominated individuals will be zero whereas a high $\Psi(x)$ value denotes that x is dominated by many individuals.

Density Estimation Technique

In addition, the density estimation technique was introduced in order to identify the preference between two individuals with the same fitness values. This tech-

nique is an adaptation of the k-th nearest neighbour method (Silverman, 1986), where the density at a given point is a decreasing function of the distance to the k-th nearest point. For each individual x, the distance to all individuals y in archive A and the population P are computed and sorted in a list. After sorting the list in an increasing order, the distance at the k-th element, ξ_x^k, is obtained and the density of individual x is defined by (Zitzler et al., 2001):

$$Đ(x) = \frac{1}{\xi_x^k + 2} \tag{3.14}$$

The value two is added in the denominator of the density function in order to ensure that its value is within the range (0,1). Finally, the fitness value of individual x is obtained by adding the density $Đ(x)$ and the raw fitness value $\Psi(x)$ as:

$$F(x) = \Psi(x) + Đ(x) \tag{3.15}$$

Archive Truncation Method

During the selection process, all the non-dominated individuals (i.e., individuals which have fitness values less than one) from the archive and the population are maintained in the archive of the next generation as follows:

$$A' = \{x | x \in P \cup A \wedge F(x) < 1\} \tag{3.16}$$

If the archive A' is not full, the best $\mathcal{NP} - \mid A' \mid$ dominated individuals in the previous population and archive are copied to the new archive. On the other hand, when the number of non-dominated individuals is more than $\mathcal{NP}$, the archive truncation process is employed based on the distance measure ξ_x^k. The individual which has the minimum distance to another individual is selected to be removed.

3.3.2.3 Pareto Envelope-based Selection Algorithm

The Pareto Envelope-based Selection Algorithm (PESA) was introduced by Corne et al. (2000). Similar to SPEA2, it uses an external population to maintain the non-dominated solutions found. In this approach, the objective space is divided into $\mathbb{k}$-dimensional cells. The density of each cell is evaluated by the number of individuals resides in the cell (Konak et al., 2006). This density measure is used to achieve diversity of the individuals in the archive.

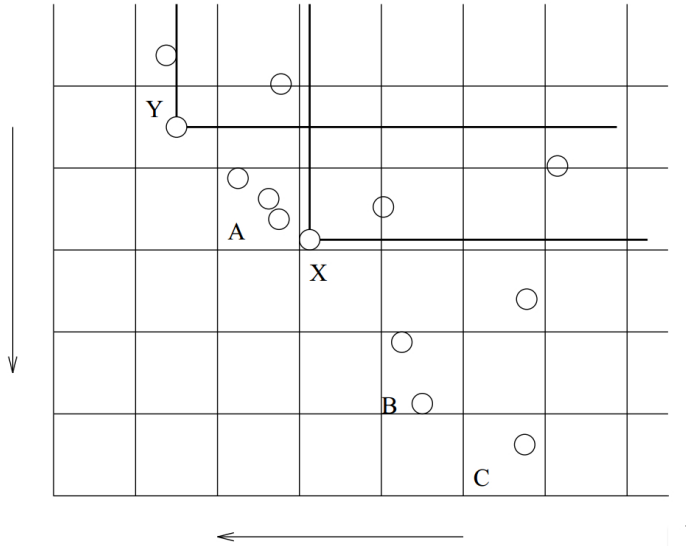


Figure 3.7: Cell-based selection method in PESA-II (Corne et al., 2001).

An improved version of the PESA is called PESA-II (Corne et al., 2001). PESA-II was proposed in order to reduce the computational cost associated with Pareto ranking (Coello et al., 2007; Corne and Knowles, 2003). In this technique, instead of assigning a selective fitness to an individual, it is assigned to the cells in the objective space which are occupied by at least one element. During the selection process, the cell with the best fitness is selected. A cell which is sparsely occupied has a higher chance to be selected than a crowded cell. For example, in Figure 3.7, hyperbox C has a better selective fitness than hyperbox B (Corne et al., 2001). Once the cell is selected, individuals within the cell are randomly chosen to employ crossover and mutation operations to the mating pool (Konak et al., 2006).

3.3.2.4 Pareto Archived Evolution Strategy

The Pareto Archived Evolution Strategy (PAES), designed by Knowles and Corne (2000), is a variant of (1+1) evolution strategy. PAES represents the simplest EA which only employs local search operations. Nonetheless, it is capable to achieve efficient solutions by maintaining an archive with previously found non-dominated solutions. The archive is exploited as a reference set in evaluating the quality of new candidate solutions. PAES also uses a novel approach to maintain diverse solutions by using a grid in the objective function space to compute the crowding distance.

Algorithm 3.9: Pseudocode for $test(p, p', A, \mathcal{A}_{size})$ (Tan et al., 2006).

Input: p: a solution,
p': a candidate solution,
A: an archive,
$\mathcal{A}_{size}$: the size of the archive A;
Output: p, p', A

```
if ( | A | < A_size) then
    add p' to the archive A;
    if (p' is in a less crowded region of A than p) then
        p ← p';
else
    if p' is in a less crowded region of A than some members in A then
        remove a member of A from the most crowded region;
        add p' to the archive A;
        if (p' is in a less crowded region of A than p) then
            p ← p';
    else
        if (p' is in a less crowded region of A than p) then
            p ← p';
```

Algorithm 3.10: PAES Procedure.

Input: $\mathcal{A}_{size}$: the size of an archive A;
Output: A;

$A := \emptyset$;
generate a random current solution p;
evaluate p and add to archive A;
repeat
    mutate p to produce a candidate p';
    evaluate p';
    if *(* $p \preceq p'$ *)* **then**
        discard p';
    else if *(*$p' \preceq p$ *)* **then**
        replace p with p';
        add p' to A;
    else if *(* $\exists\, p'' \in A \ \wedge \ p'' \preceq p'$ *)* **then**
        discard p';
    else
        apply $test(p, p', A, \mathcal{A}_{size})$ to determine which individual becomes the new current solution and whether to add p' to the archive A (see Algorithm 3.9)
until *termination criteria met*;

Algorithm 3.10 describes the procedure of PAES (Tan et al., 2006). At each generation, a candidate solution p' is generated from a single parent solution p by employing a mutation operation. Acceptance to the archive A is based on dominance criteria. Each time a candidate solution p' is generated, it is added to the archive A if it is not dominated by any members in the archive. If the archive size $\mathcal{A}_{size}$ exceeds a threshold, then it is pruned by removing the individuals that resides in the most crowded region. The crowding procedure is based on recursively dividing up the $\mathcal{D}$-dimensional objective space in 2^d equal-sized cells, where d is a predefined depth parameter. The procedure repeats until the termination criteria is met.

3.3.3 Decomposition-based MOEA

A MOEA based on decomposition (MOEA/D), proposed by Zhang and Li (2007), decomposes the multi-objective optimization problems under consideration into a number of scalar objective optimization problems (SOPs). The objective of each SOP, called a sub-problem, is a weighted aggregation of the individual objectives.

In the course of evolution, it solves all sub-problems simultaneously and each of them is optimized by making use of the information obtained from its neighbouring sub-problems. The neighbourhood relations among these sub-problems are defined based on the distances between their aggregation weight vectors. Each individual in the population is associated with a sub-problem and it is usually the best solution found so far. At each generation, a new solution for each sub-problem is generated by the recombination of a number of solutions from its neighbouring sub-problems. The current solution of the sub-problem is replaced with the new one if the latter is better. Moreover, a sub-problem also shares its newly generated solution with some (or all) of its neighbouring sub-problems which will update their current solutions if the shared solution is better. An advantage of MOEA/D is that a scaler objective local search can easily be applied in each sub-problem (Mishra et al., 2014; Zhang and Li, 2007; Zhou et al., 2011).

Since its introduction, MOEAs with decomposition have gained increasing research interests and have been applied to a wide range of applications (Chang et al., 2008; Konstantinidis and Yang, 2011; Peng et al., 2009; Zhang et al., 2010). Li and Zhang (2009) proposed a new version of MOEA/D for continuous multi-objective optimization, where DE and polynomial mutation are incorporated in order to achieve global exploration and local exploitation. Several studies had been performed by hybridizing MOEA/D with other search algorithms (Al Moubayed et al., 2010; Li and Landa-Silva, 2008, 2011). Some studies had been performed by adopting a parallel version (Durillo et al., 2011; Ishibuchi et al., 2010). More information on MOEA/D can be found at the webpage `http://dces.essex.ac.uk/staff/zhang/webofmoead.htm` maintained by Prof.

Qingfu Zhang from University of Essex.

3.3.4 Preference-based MOEAs

Depending on the preference of a decision maker, the multi-objective optimization methods can be classified into three categories: priori methods, progressive methods and posteriori methods(Miettinen, 1999). In a priori method, the decision maker express his/her preference before the search and such preference information is used to focus the search on the interested regions in the objective space. In the progressive methods, the decision maker is involved in the search process and interactively expresses his/her preference to guide the search. In a posteriori method, the well distributed efficient solutions are generated first and the decision maker then select his/her most preferred one.

Many studies have been conducted by employing the preferences in solving multi-objective optimization problems. Fonseca and Fleming (1993) introduced a prefer-ence-based MOEA where the rank of the individuals in the population is evaluated by the Pareto dominance and the decision maker's preferences. Deb et al. (2006) considered the use of reference points to include preference information. The solutions in a population are ranked by the Euclidean distance from the reference point.

Friedrich et al. (2011) proposed different models and incorporated the decision maker's preferences by weighting information on the objective space. This preference model had been successfully integrated with NSGA-II and SPEA2. Thiele et al. (2009) presented an interactive approach where the decision maker provide his/her preference as reference points. The provided reference point is used to generate a new population by combining the fitness function and an achievement scalarizing function. More information on preference-based MOEAs can be found in (Rachmawati and Srinivasan, 2006; Zhou et al., 2011).

3.3.5 Indicator-based MOEAs

The performance metrics (see Section 3.4) are primarily used to measure the quality of $\mathbb{P}_{known}$ in multi-objective optimization problems. The indicator-based MOEAs utilize these performance metrics in evolutionary selection process to guide the search. Zitzler and Künzli (2004) introduced a general indicator-based evolutionary algorithm (IBEA). IBEA compares a pair of solutions by an arbitrary metric to guide the search and does not require any additional diversity preservation mechanism. IBEA was reported to achieve better overall performance compared with NSGA-II and SPEA2.

Bader and Zitzler (2011) studied the high computational complexity of the hypervolume calculation in many objective optimization problems and proposed a fast hypervolume-based MOEA based on Monte Carlo simulations. Boonma and Suzuki (2011) presented a prospect indicator based MOEA, called PIBEA, where the proposed prospect indicator assesses the potential of each solution and helps to produce better offsprings. PIBEA is designed to solve multi-objective optimization problem efficiently by maintaining sufficient selection pressure and high level of diversity. PIBEA was reported to outperform NSGA-II, SPEA2 in terms of convergence and diversity measures. More information on preference-based MOEAs can be found in (Zhou et al., 2011).

3.4 Performance Measures for MOEAs

To evaluate the performance of the multi-objective evolutionary algorithms from various aspects, several performance metrics have been proposed in the literature (Knowles and Corne, 2002; Van Veldhuizen and Lamont, 2000; Zitzler et al., 2003). These metrics are mainly defined based on how close the obtained solutions are to the true Pareto front and how evenly the solutions are distributed along the obtained efficient frontier (Zhou et al., 2011; Zitzler et al., 2000).

In this section, we describe four metrics which are widely used by many studies to evaluate the performance of the MOEAs. Some of these metrics are used

to measure only the convergence or diversity, and others consider both criteria. Figure 3.8 shows the classification of quality indicators.

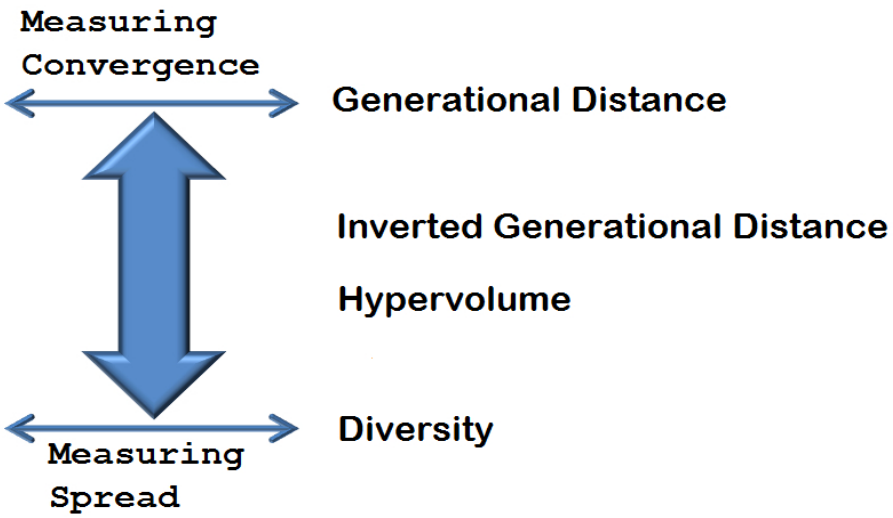


Figure 3.8: A classification of performance metrics (adapted from Durillo et al. (2011)).

3.4.1 Generational distance (GD)

The generational distance (Van Veldhuizen and Lamont, 1998) is a widely used metric to measure the convergence of an algorithm. It measures how far the solutions of the computed Pareto front obtained by an algorithm are from those in the true Pareto front.

$$GD = \frac{\sqrt{\sum_{i=1}^{\widehat{Q}} \hat{d}_i^2}}{\widehat{Q}}$$

where $\widehat{Q}$ is the number of solutions in the obtained front ($\mathbb{P}_{known}$) and $\hat{d}_i$ is the Euclidean distance (measured in objective space) between each solution in the obtained front and the *nearest* solution in the true Pareto front. The value of $GD = 0$ indicates that all the generated solutions are on the true

Pareto front (i.e., $\widehat{Q} \subseteq Q^*$). Figure 3.9, for example, shows that $\widehat{Q} = 3$, $\hat{d}_1 = \sqrt{(2.5-2)^2 + (9-8)^2}$, $\hat{d}_2 = \sqrt{(3-3)^2 + (6-6)^2}$, $\hat{d}_3 = \sqrt{(5-4)^2 + (4-4)^2}$ and $GD = 0.5$.

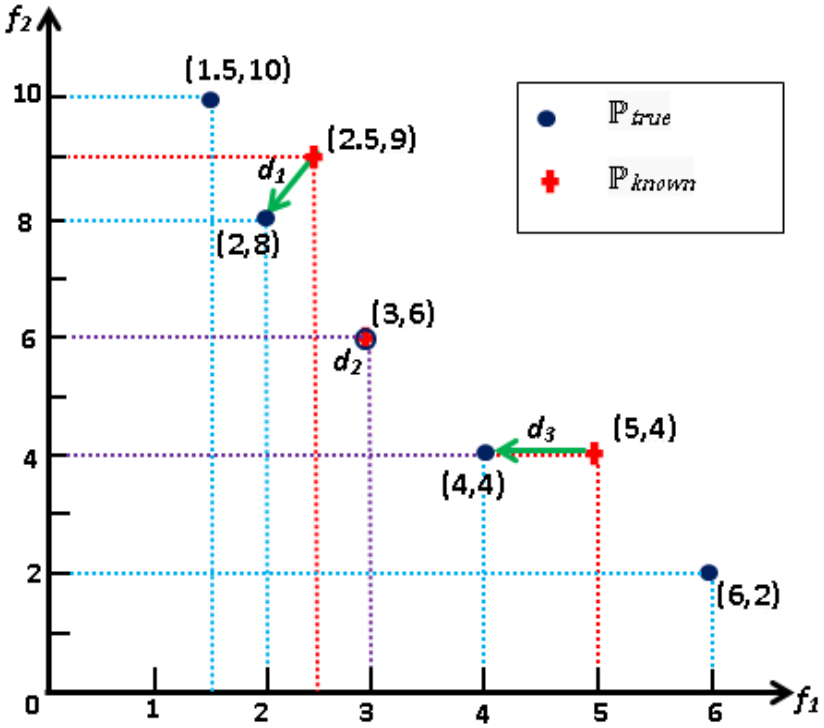


Figure 3.9: Example illustration of the generational distance (GD) metric (adapted from Coello et al. (2007)).

3.4.2 Inverted generational distance (IGD)

The inverted generational distance (Sierra and Coello Coello, 2005) uses the true Pareto front as a reference and measures the distance of each of its elements from the true Pareto front to the non-dominated front obtained by an algorithm. It is mathematically defined as:

$$IGD = \frac{\sqrt{\sum_{i=1}^{Q^*} d_i^2}}{Q^*}$$

where Q^* is the number of solutions in the true Pareto front ($\mathbb{P}_{true}$) and d_i is the Euclidean distance (measured in objective space) between each solution from $\mathbb{P}_{true}$ and the nearest member from the set of non-dominated solutions found by the algorithm ($\mathbb{P}_{known}$). This metric measures both the diversity and the convergence of an obtained non-dominated solution set. A smaller value of this metric implies a better quality of the approximation. A value of IGD equals to zero indicates that all obtained solutions lie on the true Pareto front and have the best possible spread. Figure 3.10, for example, shows that $Q^* = 5$, $d_1 = \sqrt{(1.5-2.5)^2+(10-9)^2}$, $d_2 = \sqrt{(2-2.5)^2+(8-9)^2}$, $d_3 = \sqrt{(3-3)^2+(6-6)^2}$, $d_4 = \sqrt{(4-5)^2+(4-4)^2}$, $d_5 = \sqrt{(6-5)^2+(2-4)^2}$ and $IGD = 0.6$.

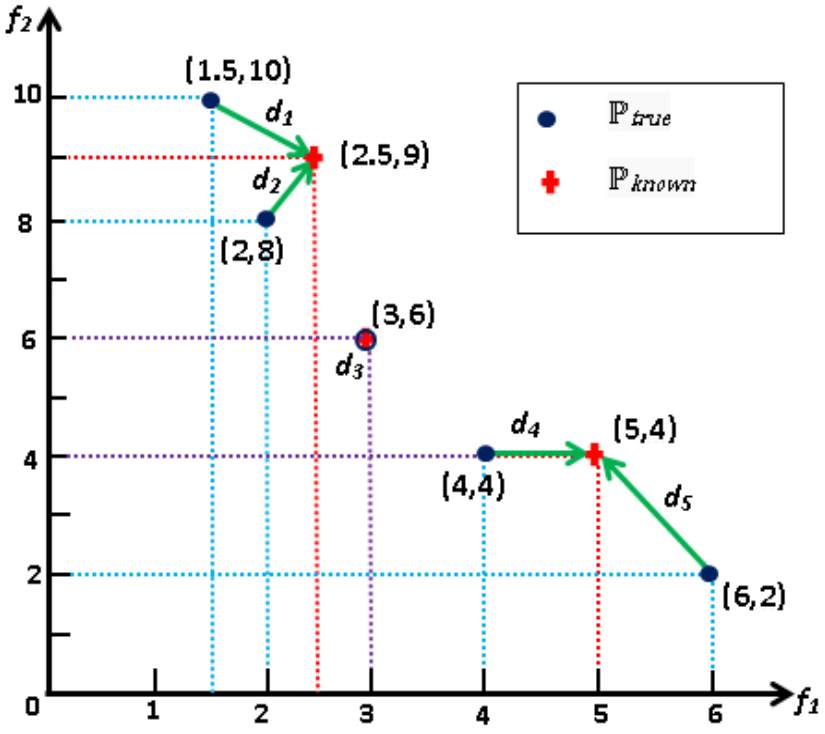


Figure 3.10: Example illustration of the inverted generational distance (IGD) metric.

3.4.3 Hypervolume (HV)

Hypervolume metric (Zitzler and Thiele, 1999), also known as S-metric or Lebesgue measure, is widely recognized as a unary value which is able to measure both closeness of the solutions to the optimal set and diversity of the obtained solutions. The hypervolume metric calculates the volume of the objective space covered by members of an obtained Pareto set $\mathbb{P}_{known}$ bounded by a reference point r. The reference point r is found by constructing a vector of worst objective function values.

Let $\widehat{Q}$ be the set of non-dominated solutions obtained by an algorithm. For each solution $p \in \widehat{Q}$, a hypercube $\mathbb{v}_p$ from solution p and the reference point r is measured. The hypervolume (HV) value is calculated by summing all hypercubes $\mathbb{v}_i$. The hypervolume (HV) is mathematically described as follows:

$$HV = volume(\bigcup_{p=1}^{|\widehat{Q}|} \mathbb{v}_p)$$

When comparing two sets of non-dominated solutions, the set which conveys a larger HV value is considered to be better both in terms of proximity and diversity. The main advantage of the hypervolume metric is that it does not depend on the prior knowledge of the true Pareto front.

Figure 3.11 shows the graphical representation of the hypervolume metric for the minimization of two objectives: f_1 and f_2. In this example, the hypervolume is represented by the grey area delimited by the non-dominated solutions ($\widehat{Q} = \{p_1, p_2, p_3, p_4, p_5\}$) and the reference point r.

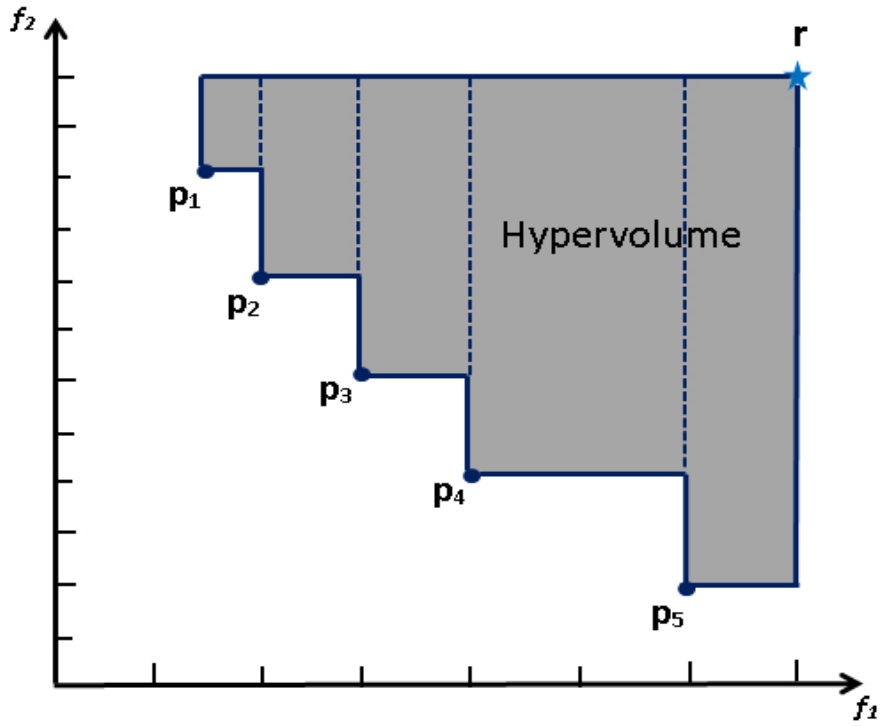


Figure 3.11: Graphical illustration of the hypervolume (HV) metric for a bi-objective minimization problem.

An accurate calculation of the hypervolume (HV) metric requires a normalized objective space and we use the linear normalization technique proposed by Knowles et al. (2006) as follows:

$$f_i = \frac{f_i - f_i^{min}}{f_i^{max} - f_i^{min}}$$

where f_i^{min} and f_i^{max} are the minimum and maximum value of the i^{th} objective. The value of f_i^{min} and f_i^{max} are set as the minimum and maximum value obtained from running all considered algorithms.

3.4.4 Diversity metric (Δ)

The diversity metric (Δ) (Deb et al., 2002) measures the performance indices of distribution and spread simultaneously for two-objective optimization problems.

In the literature, this metric is also known as spread. The diversity metric (Δ) is defined as follows:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{\widehat{Q}-1} \mid d_i - \overline{d} \mid}{d_f + d_l + (\widehat{Q} - 1)\overline{d}}$$

where $\widehat{Q}$ is the number of solutions in the obtained non-dominated front ($\mathbb{P}_{known}$), d_i is the Euclidean distance (measured in objective space) between consecutive solutions in $\mathbb{P}_{known}$, and $\bar{d}$ is the average of these distances. The parameters d_f and d_l are the Euclidean distance between the extreme solutions of the Pareto optimal front and the boundary solutions of the obtained non-dominated front $\mathbb{P}_{known}$. Figure 3.12 depicts the calculation of the diversity metric. A smaller value of the spread (Δ) indicates that the obtained non-dominated front has wider spread and more uniformly distributed along the true Pareto front.

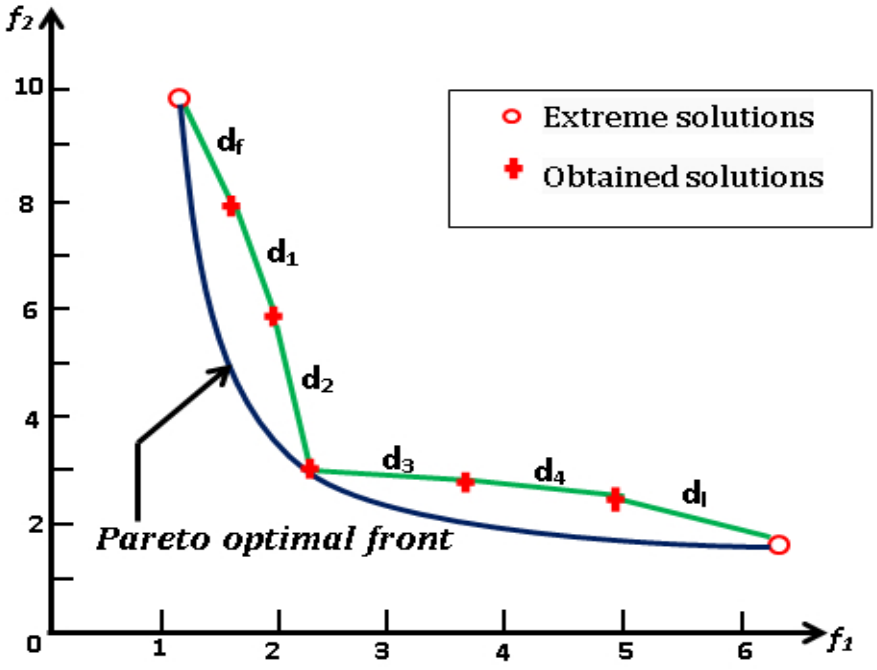


Figure 3.12: Diversity metric (Δ) (Durillo et al., 2011).

3.5 Summary

In this chapter, we review a number of well-known and widely used EAs. In the first part of the chapter, a formal definition of a multi-objective optimization problem and concept of Pareto optimality are described. In the second part of the chapter, we review several population-based evolutionary algorithms based on single objective (weighted sum) methods, Pareto-based methods, decomposition-based methods, preference-based methods and indicator-based methods for multi-objective optimization problems. Finally, the performance measures of the multi-objective evolutionary algorithms are also described.

In this thesis, we consider four variants of portfolio optimization problems with different combination of practical trading constraints and different risk measures. As noted in Section 1.1, when the basic model is extended with the cardinality constraint, the problem becomes NP-hard (Bienstock, 1996; Moral-Escudero et al., 2006; Shaw et al., 2008). Under time and resource limitations, EAs are the ideal choices for solving the portfolio optimization problems considered in this thesis. This chapter provides an overview of the different variants of evolutionary optimization techniques which may be adapted or hybridized in later chapters of this thesis.

Chapter 4

A Hybrid Algorithm for Constrained Portfolio Optimization

"Simplicity is the ultimate sophistication."

Lenonardo Da Vinci

4.1 Introduction

The basic MV model has several limitations which prohibit its use in practice (see Section 2.4). As a result, several extensions and modifications have been developed in the literature to address real-world constraints. From a practical point of view, real-world investors commonly face restrictions such as cardinality and bounding constraints. These constraints are generally imposed in order to prevent the portfolio from being composed of too many assets with small holdings. Extending the basic model with a cardinality constraint already transforms the model from a quadratic optimization model to a quadratic mixed-integer problem (QMIP), which has been proved to be *NP-hard* (Bienstock, 1996; Moral-Escudero et al., 2006; Shaw et al., 2008). Since QMIPs are hard to solve optimally, many researchers and practitioners have applied metaheuristic approaches to solve the cardinality constrained portfolio optimization problem.

In this chapter, we present a new hybrid evolutionary algorithm (PBILDE) for portfolio optimization problems. We consider the extended mean variance portfolio optimization problem with cardinality and quantity constraints (CCMV) and adopt a single objective optimization approach by aggregating the objective functions. These two constraints are the most commonly adopted ones in the literature (Metaxiotis and Liagkouras, 2012). The detailed description of the components of the hybrid algorithm is then presented. The performance of the algorithm is compared with other existing studies.

This work is motivated by the efficient exploration of DE mutation schemes and its ability to reduce the chances of being stuck in local optimum (Price et al., 2006). A review of previous works clearly indicates the trend of hybridizing DE with several evolutionary operators (Brest et al., 2006; Das and Suganthan, 2011; Pholdee and Bureerat, 2013; Sun et al., 2005; Wang et al., 2010). PBIL extracts global statistical information about the search space and exploits this information to create promising solutions. In this approach, we adopt the PBIL scheme to identify the promising assets and DE mutation scheme for efficient exploration of the proportion of assets. The main motivation behind the hybridization of PBIL and DE is to exploit the good features of different strategies in hope of achieving better performance than the individual's performance. To the best of our knowledge[4], there is no comparative study of the hybridization of DE and PBIL to portfolio optimization problems. This study is intended to fill in this gap.

[4] Pholdee and Bureerat (2013) recently presented a hybrid multi-objective evolutionary algorithm (RPBIL-DE) using continuous population based incremental learning and differential evolution for multi-objective design of trusses. Their work was published around the same time as our work (Lwin and Qu, 2013). The main difference of PBILDE from the work proposed by Pholdee and Bureerat (2013) is that we adopted a binary PBIL whereas they adopted a real-code PBIL.

4.2 The mean variance portfolio with cardinality and bounding constraints (CCMV)

In this work, Markowitz's MV model is extended with cardinality and quantity constraints and the weighted sum method (see Section 3.2.2) is adopted to model the CCMV as follows:

$$minimize \quad f(p) = \lambda \left[\sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} w_i w_j \sigma_{ij} \right] + (1-\lambda) \left[-\sum_{i=1}^{\mathcal{N}} w_i \mu_i \right] \tag{4.1}$$

$$subject\ to \quad \sum_{i=1}^{\mathcal{N}} w_i = 1, \tag{4.2}$$

$$\sum_{i=1}^{\mathcal{N}} s_i = \mathcal{K}, \tag{4.3}$$

$$\epsilon_i s_i \leq w_i \leq \delta_i s_i, i = 1, ..., \mathcal{N}, \tag{4.4}$$

$$s_i \in \{0, 1\}, i = 1, ..., \mathcal{N} \tag{4.5}$$

where $f(p)$ defines the fitness of individual p and λ is a weighting parameter between the two objectives (see Section 2.2.1), $\mathcal{K}$ is the desired number of invested assets in the portfolio, s_i denotes whether asset i is invested or not. If s_i equals to one, asset i is chosen to be invested and the proportion of capital w_i lies in $[\epsilon_i, \delta_i]$, where $0 \leq \epsilon_i \leq \delta_i \leq 1$. Otherwise, asset i is not invested and w_i equals to zero. As noted before in Section 2.4.1, cardinality constraint can be relaxed as an inequality constraints in the literature. Our focus in this work, however, is equality constraint where we seek exactly $\mathcal{K}$ assets in a portfolio.

4.3 Related Work

Since Markowitz's seminal work, many studies have been conducted on the computational techniques and recently metaheuristics for the portfolio optimization problems. In the literature, many studies have been performed by solving variants of the CCMV model (see Section 2.4.1) or portfolio optimization problems with different combinations of constraints. In this section, we only summarize the studies that include a cardinality (with equality) and quantity constraints.

Chang et al. (2000) employed three heuristic algorithms; genetic algorithm (GA), tabu search (TS) and simulated annealing (SA) to find the constrained efficient frontier. Their work also showed that the efficient frontier becomes discontinuous in the presence of cardinality restrictions. Computational results are presented for five datasets involving up to 225 assets. These datasets are made publicly available from OR-Library (Beasley, 1990). For the unconstrained problem, GA performs the best with mean percentage error close to zero. SA ranks the second and followed by TS. For the constrained problem, no single algorithm performs consistently better than the others for all five datasets. They suggest an approach that uses a pool of results from all three heuristics. Since the work of Chang et al. (2000), many studies have been performed using the same datasets which are publicly accessible.

Cura (2009) applied a particle swarm optimization (PSO) for the considered problem and compared with GA, TS and SA. This work also concluded based on computational results that none of the four heuristic algorithms has outperformed the others in all five OR-Library datasets. Deng et al. (2012) also presented an improved PSO and compared with different variants of PSO as well as three methods from Chang et al. (2000). Experimental results showed that the proposed PSO outperformed the others in most instances. Fernández and Gómez (2007) applied a heuristic method based on a Hopfield neural network to the constrained problem. Comparisons of their proposed method are performed against Chang et al. (2000) and computational results showed that no single method outperforms the others.

Moral-Escudero et al. (2006) proposed a hybrid strategy that combines a GA with quadratic programming. The proposed hybrid method uses GA to select the optimal subset of the available assets and quadratic programming to determine the proportion of capital to be invested in each asset. Two different encoding schemes and crossover operators for the GA were investigated. Computational results on five OR-Library datasets showed that subset encoding with random assorting recombination performed better than TS.

Xu et al. (2010) presented a hybrid algorithm that combine a PBIL and a continuous population based incremental learning (PBILc) to select the optimal subset of the available assets and to determine the proportion of capital to be invested in each asset respectively. The experimental results showed that the proposed algorithm was competitive to GA and PSO and achieved good results in searching efficient portfolios with high expected returns.

Mishra et al. (2014) presented a non-dominated sorting multi-objective particle swarm optimization algorithm (NS-MOPSO) and compared with four single objective evolutionary algorithms, namely GA, TS, SA and PSO, and a set of multi-objective evolutionary algorithms based on non-dominated sorting and decomposing frameworks. Experiments are performed by using six different market indices, five OR-library datasets and BSE-500 (Bombay Stock Exchange) of India. Computational results showed that the proposed approach is capable to identify good Pareto solutions, maintaining adequate diversity.

Ehrgott et al. (2004) presented a multi-criteria decision making approach. They considered five different objectives and these objectives are combined via weighted utility functions. Their work studied and compared four approaches: a two-phase local search algorithm, SA, TS and GA. Computational results showed that the GA performed well in some test problems and the two-phase local search algorithm performed well on other test problems.

4.4 A Hybrid Algorithm for CCMV

In this section, we present a new hybrid algorithm, PBILDE, to efficiently address the CCMV model described in Section 4.2. This work had been published at Applied Intelligence (Lwin and Qu, 2013).

PBILDE maintains a population of chromosomes, each representing a potential solution to the portfolio selection problem with cardinality and bounding constraints. It also maintains a real-valued probability vector to denote the probability of each asset being selected in high quality portfolios. The above stated CCMV model (see Section 4.2) can be seen as two sub-problems, the determination of the selection of assets and the allocation of capital to each asset. In each iteration of PBILDE, the probability vector is used to generate a population of solutions determining which assets are included in each solution. The DE offspring generation scheme (see Section 4.4.6) is used to allocate the proportions of assets.

PBILDE adopts elitism by maintaining an archive of the best solutions found during the evolution (see Section 4.4.3). A partially guided mutation (see Section 4.4.5) is also adopted to guide further search towards selecting favourable set of assets. The evolution process continues until a stopping criterion is met (i.e, the current best objective function value is better than a given value or it reaches to a certain number of generations). The procedure of the PBILDE is described in Algorithm 4.1.

Algorithm 4.1: PBILDE Procedure.

Input: $\mathcal{MP}$, $\mathcal{MR}$: mutation probability and mutation rate,
$\mathcal{LR}$, $\mathcal{N}\_\mathcal{LR}$: learning rate and negative learning rate,
$\mathcal{F}$: an amplification factor,
$\mathcal{NP}$: the number of population,
$\mathcal{N}$: the number of available assets,
λ: the weighting parameter in evaluating the objective function;

Output: $best$;

$A := \emptyset$; $P := \emptyset$; $best := \emptyset$; $cbest := \emptyset$, $cworst := \emptyset$;
for $i := 1$ *to* $\mathcal{N}$ **do** /* Initialization */
  $v_i := 0.5$;
for $j := 1$ *to* $\mathcal{NP}$ **do**
  $p_j \leftarrow$ randomly generate an individual;
  if *constraints are violated* **then**
    $p_j \leftarrow$ apply repair mechanism (see Section 4.4.7);
repeat
  for $j := 1$ *to* $\mathcal{NP}$ **do**
    evaluate $f(p_j)$; `// see Eq.(4.1)`
  $A \leftarrow$ maintain the $\mathcal{A}_{size}$ best portfolio(s) found so far (see Section 4.4.3);
  $best \leftarrow$ best portfolio in the archive A;
  $cbest \leftarrow$ best portfolio of the current population p;
  if $(f(cbest) > f(best))$ **then**
    $replace$ $\mathcal{A}_{size}$ worst individuals of the current population with $\mathcal{A}_{size}$ best individuals from the archive A;
  $cworst \leftarrow$ worst portfolio of the current population;
  $v \leftarrow update$ v by learning from $cbest$ and $cworst$ individuals of the current population p (see Section 4.4.4);
  $perform$ Partially Guided Mutation (see Section 4.4.5);
  `// Generate offspring (see Section 4.4.6)`
  $generate$ a trial population by DE offspring generation scheme;
  $select$ individuals of the next population using greedy selection;
until *(pre-defined number of generations or vector* v *has converged)*;

4.4.1 Solution representation and encoding

In our solution representation, two vectors of size $\mathcal{N}$ are used to define a portfolio p: a binary vector s_i, $i = 1, ..., \mathcal{N}$ denoting whether asset i is included in the portfolio, and a real-value vector w_i, $i = 1, ..., \mathcal{N}$ representing the proportions of the capital invested in the assets.

In the literature, different encodings have been proposed. The most commonly adopted encoding is to use a single real-value vector that denotes the weight allocations of the assets in a portfolio. In fact, better algorithmic performance can be achieved if a problem specific representation is utilized. Streichert et al. (2004a,b,c) investigated and compared performances of different encodings and crossover operators. They concluded that hybrid (binary+real) encoding is the best suited for the mean variance cardinality constrained portfolio selection problem. This hybrid scheme also facilitates the removal and adding of assets to portfolios. Therefore, in this work we have adopted this encoding approach.

4.4.2 Initialization

In PBILDE, the evolution is carried out on a population of a predefined number of individuals p which are represented by s_i and w_i. The probability vector v_i is used to determine if asset i is selected in a portfolio, i.e. $s_i = 1$ or $s_i = 0$. An initial population of the predetermined number of portfolios from the $\mathcal{N}$ available assets is randomly generated. Initially, the probability vector v_i is set to 0.5 to give equal chances to each asset being selected. The proportions of the selected assets in each solution are then randomly generated from the given lower and upper bounds by adopting Gaussian distribution. The randomly constructed portfolio could violate the constraints in the model and the constraint handling scheme described in Section 4.4.7 is applied to adjust and normalize the weights (See Fig. 4.1).

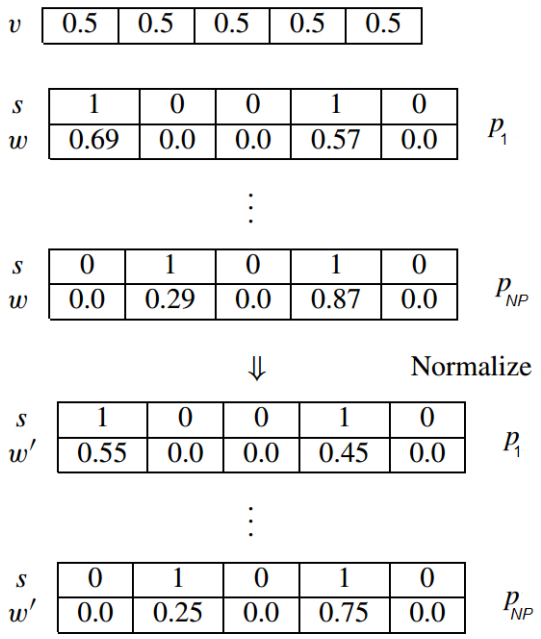


Figure 4.1: Example of an initial population and probability vector.

4.4.3 Maintaining the Archive

During the evolution, an archive A reserves $\mathcal{A}_{size}$ best portfolios. At each iteration during the evolution, the archive is updated to maintain the best individuals found so far. If the best individual ($cbest$) in the new sampled population at the current generation is worse than the global best individual ($best$) found so far, then the $\mathcal{A}_{size}$ worst individual(s) of the current population are replaced by the $\mathcal{A}_{size}$ global best individual(s) from the archive. This strategy is incorporated to promote the convergence of the algorithm. The motive to maintain the archive A is to prevent the loss of the global best solutions found during the search as well as to exploit those best solution(s) to help generate better solutions.

4.4.4 Updating the probability vector

In PBILDE, the probability vector v is used to store statistic information collected during the evolution to guide the generation of the following populations. At

each generation, the learning rate ($\mathcal{LR}$) and negative learning rate ($\mathcal{N}\_\mathcal{LR}$) are used to update the probability vector (υ). They control not only the speed at which the probability vector is shifted to resemble the best solution vector but also the portion of the search space that will be explored (Folly and Venayagamoorthy, 2009; Shapiro, 2003). The probability vector is updated by learning from the best solution of the current population s_i^{cbest} at a learning rate $\mathcal{LR}$ as follows:

$$\upsilon_i = \upsilon_i \times (1 - \mathcal{LR}) + s_i^{cbest} \times \mathcal{LR} \tag{4.6}$$

In addition, after the probability vector is updated at the learning rate $\mathcal{LR}$, if the i^{th} asset is selected in the best solution but it is not selected in the worst solution or vice versa (i.e, $s_i^{cbest} \neq s_i^{cworst}$) then the i^{th} asset has a higher probability of being selected than not selected. Hence, the probability vector is updated by a negative learning rate $\mathcal{N}$–$\mathcal{LR}$ in order to move away from bad solutions, i.e. learn from the bad individuals. When $s_i^{cbest} \neq s_i^{cworst}$, it is updated in the same way as PBIL in (Xu et al., 2010) as follows:

$$\upsilon_i = \upsilon_i \times (1 - \mathcal{N}\_\mathcal{LR}) + s_i^{cbest} \times \mathcal{N}\_\mathcal{LR} \tag{4.7}$$

4.4.5 Mutation of the probability vector

One of the factors to consider in designing the model in the population-based approach is to find an effective way to generate offsprings. The approximate optimality principle (Glover and Laguna, 1998) assumes that good solutions tend to have similar structure. This assumption is reasonable for many real-world problems. Based on this assumption, an ideal offspring generator aims to produce a solution which is close to the best solutions found so far in the hope that the resultant solution will not be far from the best solution and fall into a promising area of the search space (Zhang et al., 2005).

At each iteration of the evolution, each dimension of the probability vector (υ) is updated according to a certain mutation probability ($\mathcal{MP}$). By taking into account the balance between the exploitation and exploration of the search space, we adopt a new partially guided mutation. It gives an equal chance to mutate the probability vector (υ) either randomly at a mutation rate $\mathcal{MR}$ (i.e., guided mutation) or based on the global best solution. The aim is to strike a balance between exploiting good structures in the best solution and exploring other area of the search space. The pseudocode of the guided mutation is described in Algorithm 4.2.

Algorithm 4.2: Partially Guided Mutation.

Input: $cbest$: best solution in current generation,
$\mathcal{MR}$: mutation rate,
$\mathcal{MP}$: mutation probability,
υ: probability vector,
$\mathcal{N}$: the number of available assets;

Output: υ;

```
for i := 1 to N do
    if rand(0,1] < MP then
        if rand(0,1] < 0.5 then
            r := randint[0,1];
            υ_i := υ_i × (1 − MR) + r × MR;
        else
            υ_i := s_i^cbest;
```

In PBILDE, the probability vector (υ) in the main evaluation is maintained by the update and mutation based on the best and worst individuals in the population. It is then utilized to influence the selection of assets in the next generation of portfolios. The proportion of the asset is generated by DE offspring generation scheme, as explained in the following section.

4.4.6 DE Offspring Generation

The offspring generation scheme in PBILDE works with a population of solutions evolved during evolutions. The population of the next generation, P^{g+1}, is created based on the current population of the generation P^g with $\mathcal{NP}$ individuals (portfolios). It first generates a trial population $\overline{P}^{g+1}$. Each individual trial portfolio $\overline{p}_j^{g+1}$ contains two vectors:

$$\begin{aligned} &\overline{w}_{j,i}^{g+1},\ j \in \{1, ..., \mathcal{NP}\}; i \in \{1, ..., \mathcal{N}\} \\ &\overline{s}_{j,i}^{g+1},\ j \in \{1, ..., \mathcal{NP}\}; i \in \{1, ..., \mathcal{N}\} \end{aligned} \tag{4.8}$$

where $\overline{w}_{j,i}$ denotes the proportion of the i^{th} asset in the j^{th} portfolio and $\overline{s}_{j,i}$ denotes whether the i^{th} asset in the j^{th} portfolio is selected or not.

A trial population is generated as described in Algorithm 4.3. For each trial portfolio, if the i^{th} asset is selected then the weights of i^{th} asset is generated by the mutation and crossover operations. Firstly, three mutually different indexes, r_1, r_2 and r_3, which are also different from the index j of the current trial portfolio $\overline{p}_j^{g+1}$, are randomly selected from the parent population. The indexes r_1, r_2 and r_3 are randomly selected for each trial vector in the trial population.

In the mutation operation, the difference between two of the randomly selected vectors (r_1 and r_2) from the current population is multiplied by an amplification factor, $\mathcal{F}$, and it is added to the third randomly selected vector (r_3) from the current population.

The binary crossover is performed to yield the trial vector. The crossover probability $\mathcal{CR}$ represents the probability of mutating the value of the trial vector. The condition $i == r'$ is to ensure that at least one element of the trial vector is different compared to the elements of the parent vector from the current generation. Similar to the initialization process, if the trial solution generated violate the constraints in the model, the constraint handling scheme (see Section 4.4.7) is applied.

Algorithm 4.3: Procedure of Generating Trial Population

Input: $\mathcal{NP}$: the number of population,
$\mathcal{CR}$: crossover rate,
$\mathcal{F}$: an amplification factor,
$\mathcal{N}$: the number of available assets,
P^g: current population;
Output: $\overline{P}^{g+1}$: trial population;

for $j := 1$ *to* $\mathcal{NP}$ **do**
  $r' := randint[1, \mathcal{N}]$;
  for $i := 1$ *to* $\mathcal{N}$ **do**
    randomly select $r_1, r_2, r_3 \in \{1, ..., \mathcal{NP}\}$, $r_1 \neq r_2 \neq r_3 \neq j$;
    if $rand(0, 1] < v_i$ **then**
      $\overline{s}_{j,i}^{g+1} := 1$;
      if $rand(0, 1] < \mathcal{CR} \ \vee \ i == r'$ **then**
        $\overline{w}_{j,i}^{g+1} := w_{r_3,i}^{g} + \mathcal{F} \times (w_{r_1,i}^{g} - w_{r_2,i}^{g})$;
      else
        $\overline{w}_{j,i}^{g+1} := w_{j,i}^{g}$;

The population of the next generation P^{g+1} is selected from the current population P^g and the trial population $\overline{P}^{g+1}$. Each individual of the trial population is compared with the corresponding individual of the current population. PBILDE adopts the greedy selection in DE (Storn and Price, 1997). Under the greedy criterion, the better individual with the better fitness value becomes a member individual of the next generation's population:

$$p_j^{g+1} = \begin{cases} \overline{p}_j^{g+1} & \textbf{if } f(\overline{p}_j^{g+1}) < f(p_j^g) \\ p_j^g & \textbf{otherwise} \end{cases} \tag{4.9}$$

4.4.7 Constraint Handling

During the population sampling, each constructed individual must be repaired if the representative portfolio does not satisfy the constraints of the problem. If the number of the selected assets is smaller or larger than $\mathcal{K}$, then a repair operator selects or deletes an asset by using a heuristic which prioritizes the assets (Cura, 2009). The priority value a_i of each asset is defined as follows:

$$\begin{aligned}
&\Upsilon_i = 1 + (1 - \lambda) \times \mu_i, \quad i = 1, \ldots, \mathcal{N}, \\
&\Lambda_i = 1 + \lambda \times (\sum_{j=1}^{\mathcal{N}} \sigma_{ij}/\mathcal{N}) \quad i = 1, \ldots, \mathcal{N}, \\
&\Theta = -1 \times min(0, \Upsilon_1, \ldots, \Upsilon_{\mathcal{N}}), \\
&\mho = -1 \times min(0, \Lambda_1, \ldots, \Lambda_{\mathcal{N}}, \\
&a_i = \frac{\Upsilon_i + \Theta}{\Lambda_i + \mho}, \quad i = 1, \ldots, \mathcal{N}.
\end{aligned} \tag{4.10}$$

For a given asset, the priority value a_i denotes the proportion between the mean return and mean risk with respect to aversion parameter λ. The priority value a_i is used to determine which asset may be added or removed. In the case that the number of selected assets is larger, the excess assets which need to be removed are identified either randomly or by selecting those assets which have the minimum a_i values. Similarly, in the case that the number of selected assets is smaller, the new assets which need to be added are identified either randomly or by selecting those assets which have the maximum a_i values.

The budget constraint in Eq. (4.2) is satisfied by firstly normalizing the weights: $w_i = w_i / \sum_{j=1}^{\mathcal{N}} w_j$ over those assets selected based on the probability vector v. Moreover, the bounding constraint in Eq. (4.4) requires the proportion of asset i to be in the range $[\epsilon_i, \delta_i]$. If the proportion of asset after the normalization violates the upper or lower bound constraints, then it is adjusted as follows:

$$w_i = \begin{cases} w_i + \psi \times (\theta_i/\delta^*) & \textbf{if } \delta_i > w_i \\ \delta_i & \textbf{if } \delta_i < w_i \\ w_i - \phi \times (\varphi_i/\epsilon^*) & \textbf{if } w_i > \epsilon_i \\ \epsilon_i & \textbf{if } w_i < \epsilon_i, \end{cases}$$

$$\begin{aligned} \theta_i &= \delta_i - w_i, \\ \varphi_i &= w_i - \epsilon_i, \\ \delta^* &= \sum_{i=1}^{\mathcal{N}} \theta_i \qquad \textbf{where } \theta_i > 0, \\ \psi &= \sum_{i=1}^{\mathcal{N}} \mid \theta_i \mid \qquad \textbf{where } \theta_i < 0, \\ \epsilon^* &= \sum_{i=1}^{\mathcal{N}} \varphi_i \qquad \textbf{where } \varphi_i > 0, \\ \phi &= \sum_{i=1}^{\mathcal{N}} \mid \varphi_i \mid \qquad \textbf{where } \varphi_i < 0. \end{aligned} \tag{4.11}$$

The same repair strategies have been used in the literature (Chang et al., 2000; Cura, 2009; Xu et al., 2010) to adjust the number of assets and the weight of assets in the portfolio. We adopt these strategies to conduct a *fair* comparison for the computational results in the next section.

4.5 Computational Results

In this section, we describe the experiments performed and present computational results on both unconstrained and constrained portfolio optimization problem. The proposed PBILDE hybrid algorithm described in Section 4.4 has been firstly compared to two other approaches, DE and PBIL.

The DE approach differs from PBILDE in such a way that it performs selection of assets randomly before determining the proportions of assets in the weight

vector. In other words, instead of using the probability vector, it makes no effort to learn from the population in order to decide which assets are favourable to be included.

The PBIL approach adopted in our experiment was originally proposed by Xu et al. (2010). They proposed a hybrid algorithm called PBIL_CCPS by integrating a PBIL and a continuous PBIL for the CCMV. It first builds a probabilistic model about the distribution of good individuals in the search space and then samples a new generation of population using the probabilistic model. It maintains three vectors, a probability vector, a mean vector and a standard deviation vector, to learn from the previous generation. Like PBIL in (Xu et al., 2010), our adapted PBIL uses the same three vectors, probability vector, the mean and standard deviation vectors, and allocates a random proportion for the selected asset by Gaussian distribution. Unlike Xu et al. (2010), our PBIL approach with the archive of the best individuals (the elite) replaces the $\mathcal{A}_{size}$ worst solutions of the current population with the $\mathcal{A}_{size}$ global best solutions. Moreover, we introduce a partially guided mutation to exploit the information obtained during the evolution about the search space.

All three algorithms (PBILDE, PBIL and DE) in our study are applied with the elitism and partially guided mutation to demonstrate the effectiveness and efficiency of the hybrid PBILDE against the PBIL and DE with the same settings.

The proposed PBILDE has also been compared to a number of state-of-the-art approaches in the literature using the same evaluation methods to demonstrate the effectiveness of the hybrid algorithm for both the constrained and unconstrained portfolio problems. All of our experiments are coded in C# and run on a core2duo with a 2.79GHz processor and 2GB RAM. The experimental results obtained for each algorithm are the average of **20** runs. We aim to compare our work with a number of existing studies. In order to conduct a fair comparison, we perform the same number of runs.

4.5.1 Datasets

Five problem instances (D1 − D5) from OR-library (Beasley, 1990, 1999) are used to compare the performance of the tested algorithms (see Section 2.5). These datasets contain the estimated returns and the covariance matrix of five different stock market indices: Hand Seng in Hong Kong, DAX 100 in Germany, FTSE 100 in UK, S&P 100 in USA and Nikkei 225 in Japan (see Table B.2 for a small example dataset). For each set of the test data, the number of assets $\mathcal{N}$ is 31, 85, 89, 98 and 225, respectively (see Table 2.1). In the current literature of portfolio optimization problem, this set of dataset has been widely adopted and tested, and is recognized as the benchmark to evaluate computational algorithms.

4.5.2 Parameter Settings

In the parameter settings, the value of λ in the objective function Eq. (4.1) is set as $\lambda_i = (i-1)/49$ where $i = 1, 2, ..., 50$. For each value of λ, each algorithm carried out in total $1000\mathcal{N}$ fitness evaluations excluding the initializations.

Unconstrained Problem (UP): $\mathcal{K} = \mathcal{N}$, $\epsilon_i = 0$, $\delta_i = 1$ $(i = i, ..., \mathcal{N})$

Constrained Problem (CP): $\mathcal{K} = 10$, $\epsilon_i = 0.01$, $\delta_i = 1$ $(i = i, ..., \mathcal{N})$

We aim to compare our work with a number of existing studies. Therefore, the above settings are used in order to conduct a fair comparison with other existing works (see Section 4.5.3).

Initially, most parameter values of algorithms considered in this work are set by the values recommended by Xu et al. (2010) and Winker et al. (2011). Preliminary tests are then conducted to tune the parameter values of the algorithms. Table 4.1 shows the parameter values of the algorithms considered in this work.

	PBILDE	DE	PBIL
$\mathcal{A}_{size}$	$\mathcal{NP}/4$	$\mathcal{NP}/4$	$\mathcal{NP}/4$
$\mathcal{CR}$	0.8	0.8	-
$\mathcal{F}$	0.9	0.9	-
$\mathcal{LR}$	0.1	-	0.1
$\mathcal{MP}$	$1/\mathcal{N}$	-	0.05
$\mathcal{MR}$	0.05	-	0.05
$\mathcal{N}\_\mathcal{LR}$	0.075	-	0.075
$\mathcal{NP}$ for **UP**	20	20	20
$\mathcal{NP}$ for **CP**	$\mathcal{N}/4$	20	20
$\mathcal{PLR}$	-	-	[0.05,0.4]

Table 4.1: Parameter settings of PBILDE, DE and PBIL.

4.5.3 Performance Evaluation

To evaluate the performance of the algorithms, we compare the efficient frontier obtained by each algorithm with the optimal solutions provided by OR-library (Beasley, 1990, 1999). We adopt the same approach as previously used by Chang et al. (2000) to calculate the percentage deviation of each portfolio. It is evaluated by measuring the distance of the obtained efficient portfolio from the optimal efficient frontier.

As mentioned in Section 4.5.2, 50 weighting parameter (λ) values are used to calculate the efficient frontier of the portfolio selection problem (see Eq. (4.1)). We maintain a set V which consists of the best solution found for each λ. Each portfolio in set V is used to evaluate the percentage deviation from the optimal efficient frontier for the unconstrained problem.

For the constrained problems, Chang et al. (2000) considered that it is insufficient to use only set V to evaluate the performance of the algorithms. Another

set H is thus defined to store all efficient portfolios during the evolution. For each value of λ, let $p(\lambda)$ be the current best portfolio found by the algorithm. During the course of iteration, a newly found portfolio is added to H if it is better than $p(\lambda)$. Those portfolios which are dominated by other portfolio in the set are then removed from the set H. The resulting set H and set V are used to calculate percentage deviation errors for the constrained problem.

Each obtained portfolio in the set H and set V is evaluated by measuring its distance (i.e., horizontally and vertically) from the optimal unconstrained efficient frontier (UCEF). The horizontal distance (x) from the efficient frontier is measured by considering the portfolio with fixed expected return. Similarly, the vertical distance (y) from the efficient frontier is measured by considering the portfolio with fixed risk. The final percentage deviation error is then measured by taking the minimum of these two values.

4.5.4 Experimental Results

4.5.4.1 Results of Unconstrained Problems

The basic MV model can be solved efficiently by the critical line algorithm (CLA) (Markowitz, 1956; Niedermayer and Niedermayer, 2010) as well as simplex algorithm (Wolfe, 1959). The optimal solutions for the constrained problem, however, are not known. By testing on the unconstrained problems, results can be compared with the benchmark optimal solutions as a preliminary experiment. Chang et al. (2000) reasoned that an algorithm is unlikely to be capable to perform well on constrained problem unless it performs well on unconstrained problem. Therefore, the effectiveness of the algorithms are initially tested on the unconstrained problems.

Table 4.2 provides the comparison on the results of set V of three algorithms, namely PBILDE, DE and PBIL. PBILDE performed the best and obtained better results on 4 out of 5 datasets. We can conclude from the results that PBILDE is an efficient algorithm. DE is the second best in three algorithms.

Index	$\mathcal{N}$		PBILDE V	DE V	PBIL V
Hang Seng	31	MPE(%)	**0.0002**	0.0280	0.2385
		MedPE(%)	**2.63E-06**	2.81E-06	0.0257
		Time(s)	109	**105**	134
DAX 100	85	MPE(%)	**0.0052**	0.0089	1.1849
		MedPE(%)	**2.11E-05**	2.15E-05	0.4292
		Time(s)	**1445**	1522	2103
FTSE 100	89	MPE(%)	0.0059	**0.0049**	0.9813
		MedPE(%)	2.11E-06	**1.98E-06**	0.0799
		Time(s)	**1643**	1898	2145
S&P 100	98	MPE(%)	**0.0078**	0.0094	1.2361
		MedPE(%)	**3.54E-06**	3.72E-06	0.1443
		Time(s)	**2094**	2479	2700
Nikkei	225	MPE(%)	0.2733	**0.2503**	3.7411
		MedPE(%)	**2.25E-05**	2.61E-05	2.0514
		Time(s)	**24823**	28795	31903

Table 4.2: Comparison results of PBILDE with DE and PBIL for the unconstrained problem.

We also compare PBILDE with the results from Chang et al. (2000) and Xu et al. (2010) in Table 4.3, where *MedPE* and *MPE* denote the average values of the obtained median percentage error (MedPE) and mean percentage error (MPE) of set V in 20 runs. By allocating the same number of evaluations and runs, the performance of PBILDE is compared against the existing work. The comparison results show that PBILDE can achieve better solution in most instances.

			PBILDE	Chang-GA	Chang-TS	Chang-SA	Xu-GA	Xu-PSO	Xu-PBIL
Index	$\mathcal{N}$		V	V	V	V	V	V	V
Hang Seng	31	MPE(%)	**0.0002**	0.0202	0.8973	0.1129	0.0191	0.1422	0.0003
		MedPE(%)	**2.63E-06**	0.0165	1.0718	0.016	0.0166	1.07E-05	1.24E-05
DAX 100	85	MPE(%)	0.0052	0.0136	3.5645	0.0394	0.035	1.1044	**0.0023**
		MedPE(%)	**2.11E-05**	0.0123	2.7816	0.0033	0.0124	4.77E-5	3.51E-05
FTSE 100	89	MPE(%)	**0.0059**	0.0063	3.2731	0.2012	0.0109	1.143	0.0186
		MedPE(%)	**2.11E-06**	0.0029	3.0238	0.0426	0.002	0.0084	2.45E-05
S&P 100	98	MPE(%)	**0.0078**	0.0084	4.428	0.2158	0.043	2.0249	0.0137
		MedPE(%)	**3.54E-06**	0.0085	4.278	0.0142	0.0085	0.5133	2.85E-05
Nikkei	225	MPE(%)	0.2733	**0.0085**	15.9163	1.7681	0.3715	8.1781	0.0606
		MedPE(%)	**2.25E-05**	0.0084	14.2668	0.8107	0.0068	4.7023	2.69E-05

Table 4.3: Comparison results of PBILDE with Chang et al. (2000) and Xu et al. (2010) for the unconstrained problem.

4.5.4.2 Results of Constrained Problems

In this section, we outline a number of tests performed in order to decide the value of population size assignment and to evaluate the effectiveness of the new partially guided mutation and elitist scheme in PBILDE. Firstly, different population sizes are tested for the constrained problem and the results are shown in Table 4.4. Unlike for the unconstrained problem where the setting of population size does not lead to different performance, results show that for constrained problem, setting population size ($\mathcal{NP}$) as $\mathcal{N}/4$ is better than both 20 and $2\mathcal{N}$. It obtains more efficient points in set H at a much higher computation time.

Index	$\mathcal{N}$		$\mathcal{NP} = 20$		$\mathcal{NP} = 2\mathcal{N}$		$\mathcal{NP} = \mathcal{N}/4$	
			V	H	V	H	V	H
Hang Seng	31	avg MPE(%)	1.1235	0.8865	**1.1101**	0.8925	1.1431	**0.6196**
		avg MedPE(%)	1.2283	1.1050	**1.2230**	1.1060	1.2390	**0.4712**
		Number of EF points	2923		2165		**6367**	
		Time(s)	**60**		99		113	
DAX 100	85	avg MPE(%)	2.4481	1.7449	**2.4101**	1.6597	2.4251	**1.5433**
		avg MedPE(%)	2.5922	1.4291	**2.5866**	1.3945	**2.5866**	**1.0986**
		Number of EF points	3347		2021		**3378**	
		Time(s)	**526**		818		1358	
FTSE 100	89	avg MPE(%)	1.0322	1.0177	**0.9460**	**0.7204**	0.9706	0.8234
		avg MedPE(%)	1.0841	0.5443	**1.0840**	0.5203	**1.0840**	**0.5134**
		Number of EF points	2919		1574		**2957**	
		Time(s)	**590**		962		1496	
S&P 100	98	avg MPE(%)	1.9144	1.7338	**1.5688**	**1.2380**	1.6386	1.3902
		avg MedPE(%)	1.1617	0.8556	**1.1594**	0.9085	1.1692	**0.7303**
		Number of EF points	4546		2608		**4570**	
		Time(s)	**762**		1014		1901	
Nikkei	225	avg MPE(%)	0.6314	0.5198	0.5995	0.4604	**0.5972**	**0.3996**
		avg MedPE(%)	0.6017	0.5233	0.5903	0.5262	**0.5896**	**0.4619**
		Number of EF points	3967		2560		**4000**	
		Time(s)	**4955**		8070		14918	
Average		avg MPE(%)	1.4299	1.1805	**1.3269**	0.9942	1.3549	**0.9552**
		avg MedPE(%)	1.3336	0.8914	**1.3287**	0.8911	1.3337	**0.6551**

Table 4.4: Comparison results of PBILDE with different population size ($\mathcal{NP}$) for the constrained problem.

			PBILDE-with PGM		PBILDE-without PGM	
Index	$\mathcal{N}$		V	H	V	H
Hang Seng	31	MPE(%)	**1.1431**	**0.6196**	1.1444	0.7609
		MedPE(%)	**1.2390**	**0.4712**	1.2402	0.7284
		Number of EF points	**6367**		6215	
		Time(s)	113		**111**	
DAX 100	85	MPE(%)	**2.4251**	**1.5433**	2.4701	1.7668
		MedPE(%)	**2.5866**	**1.0986**	2.6003	1.4315
		Number of EF points	**3378**		3321	
		Time(s)	1358		**1332**	
FTSE 100	89	MPE(%)	**0.9706**	**0.8234**	1.0431	1.0258
		MedPE(%)	**1.0840**	**0.5134**	1.0841	0.5213
		Number of EF points	**2957**		2937	
		Time(s)	1496		**1453**	
S&P 100	98	MPE(%)	**1.6386**	**1.3902**	1.8451	1.7740
		MedPE(%)	**1.1692**	**0.7303**	1.1595	0.8161
		Number of EF points	**4570**		4240	
		Time(s)	1901		**1822**	
Nikkei	225	MPE(%)	**0.5972**	**0.3996**	0.6142	0.4476
		MedPE(%)	**0.5896**	**0.4619**	0.5965	0.4959
		Number of EF points	**4000**		3832	
		Time(s)	14918		**14327**	

Table 4.5: Comparison results of PBILDE with and without partially guided mutation.

The effectiveness of the partially guided mutation (PGM) in PBILDE is also tested and the results are shown in Table 4.5. It is clear from Table 4.5 that adopting the partially guided mutation in PBILDE contributes to better solution quality. We also tested the contribution of elitist strategy in PBILDE. The proposed elitist strategy makes use of a set of global best solutions to inject into the current pop-

ulation as a replacement with its worst members when the current best solution is worse than the global best solution. This strategy is introduced in order to reduce the chances of being stuck in the local optima. Given the results shown in Table 4.6, we would conclude that it is an advantage to maintain the archive scheme in PBILDE.

			PBILDE-with elitism		PBILDE-without elitism	
Index	$\mathcal{N}$		V	H	V	H
Hang Seng	31	MPE(%)	**1.1431**	**0.6196**	1.1241	0.7521
		MedPE(%)	**1.2390**	**0.4712**	1.2410	0.7612
		Number of EF points		**6367**		6215
		Time(s)		113		**102**
DAX 100	85	MPE(%)	**2.4251**	**1.5433**	2.4989	1.7300
		MedPE(%)	**2.5866**	**1.0986**	2.6026	1.2384
		Number of EF points		**3378**		2817
		Time(s)		1358		**1232**
FTSE 100	89	MPE(%)	**0.9706**	**0.8234**	1.0515	1.1300
		MedPE(%)	**1.0840**	**0.5134**	1.0841	0.5500
		Number of EF points		**2957**		2790
		Time(s)		1496		**1333**
S&P 100	98	MPE(%)	**1.6386**	**1.3902**	1.7889	1.7387
		MedPE(%)	**1.1692**	**0.7303**	1.1609	0.8343
		Number of EF points		**4570**		4177
		Time(s)		1901		**1702**
Nikkei	225	MPE(%)	**0.5972**	**0.3996**	0.6125	0.4480
		MedPE(%)	**0.5896**	**0.4619**	0.5961	0.4930
		Number of EF points		**4000**		3927
		Time(s)		14918		**11735**

Table 4.6: Comparison results of PBILDE with and without elitism.

Index	$\mathcal{N}$		PBILDE V	PBILDE H	DE V	DE H	PBIL V	PBIL H
Hang Seng	31	MPE(%)	**1.1431**	**0.6196**	1.2150	1.1932	1.3894	1.3737
		MedPE(%)	1.2390	**0.4712**	**1.2331**	1.2807	1.5780	1.5267
		Time(s)	113		**79**		95	
DAX 100	85	MPE(%)	**2.4251**	**1.5433**	3.3077	2.9670	2.5129	2.9245
		MedPE(%)	2.5866	**1.0986**	2.7410	2.5293	**2.5850**	2.6648
		Time(s)	1358		**1274**		1478	
FTSE 100	89	MPE(%)	**0.9706**	**0.8234**	1.3651	1.6203	1.3190	2.0282
		MedPE(%)	**1.0840**	**0.5134**	1.0975	0.9832	1.1204	1.2599
		Time(s)	**1496**		1542		1589	
S&P 100	98	MPE(%)	**1.6386**	**1.3902**	3.2008	3.2170	2.4722	3.1763
		MedPE(%)	**1.1692**	**0.7303**	1.5970	1.4973	1.2096	1.3810
		Time(s)	**1901**		1943		1992	
Nikkei	225	MPE(%)	**0.5972**	**0.3996**	1.8934	2.2053	0.7554	0.8086
		MedPE(%)	**0.5896**	**0.4619**	1.6428	1.7624	0.6592	0.6864
		Time(s)	**14918**		18327		24806	
Average		avg MPE(%)	**1.3549**	**0.9552**	2.1964	2.2406	1.6898	2.0623
		avg MedPE(%)	**1.3337**	**0.6551**	1.6623	1.6106	1.4304	1.5038

Table 4.7: Comparison results of PBILDE with population size ($\mathcal{NP}$) $= \mathcal{N}/4$ against DE and PBIL for the constrained problem.

Table 4.7 provides the comparison results of PBILDE, PBIL and DE with population size $\mathcal{NP} = \mathcal{N}/4$. PBILDE outperforms the others in all instances. Results show that PBILDE uses up less CPU time on larger problems compared to PBIL and DE. Furthermore, the lack of consideration on an efficient selection of assets in DE penalizes the algorithm performance. Both PBIL and PBILDE use a probability vector in determining the selection of assets in a portfolio. Experimental results of PBIL compared with PBILDE show that the use of the probabilistic model with the mean and standard deviation vectors in determining the proportions of the assets is not as effective as employing the DE within PBILDE. Figure-4.2 shows the comparison of the efficient frontiers of PBILDE, PBIL and DE for the constrained problem.

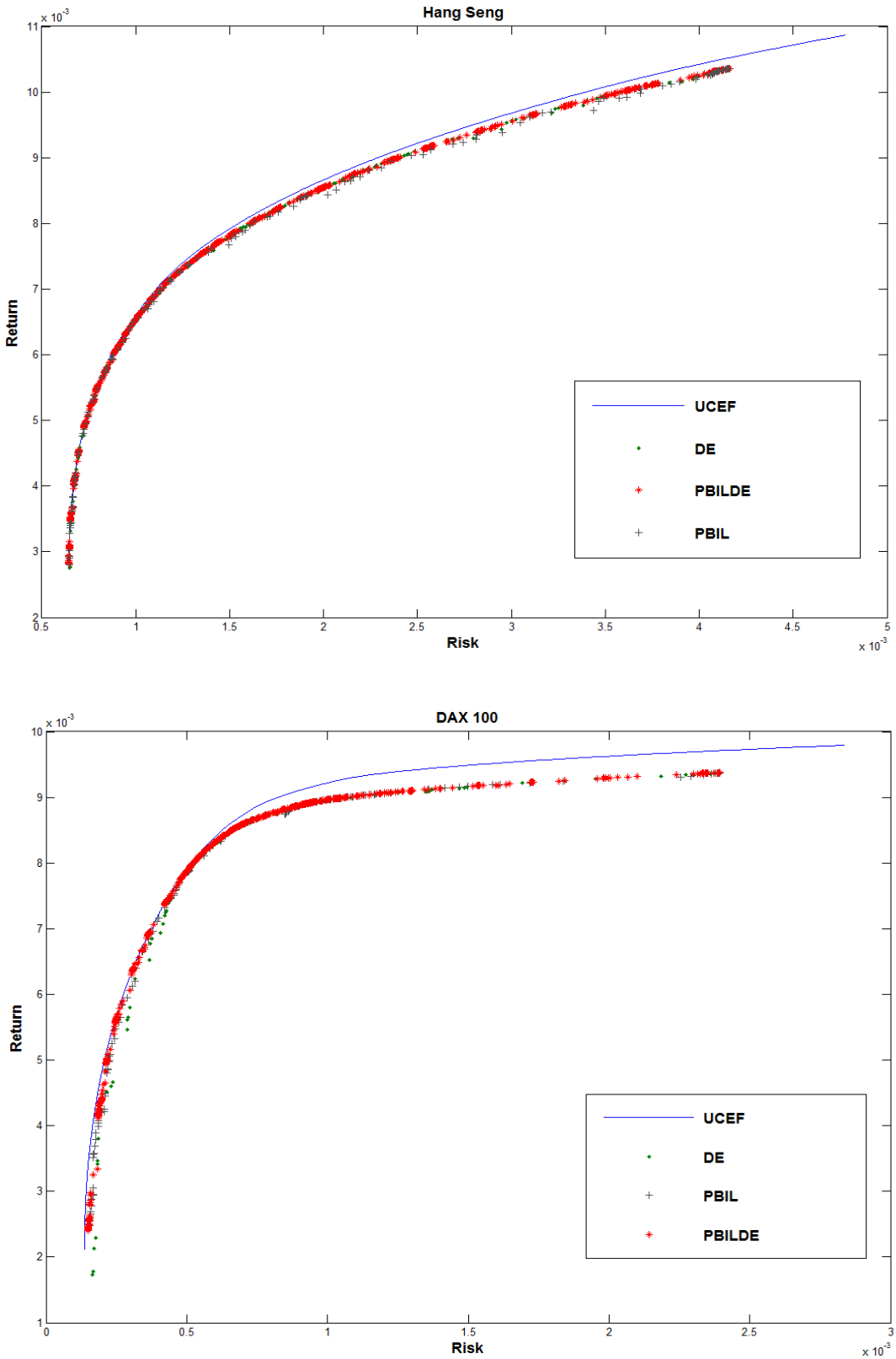


Figure 4.2: Comparison of heuristic efficient frontiers for constrained problem.

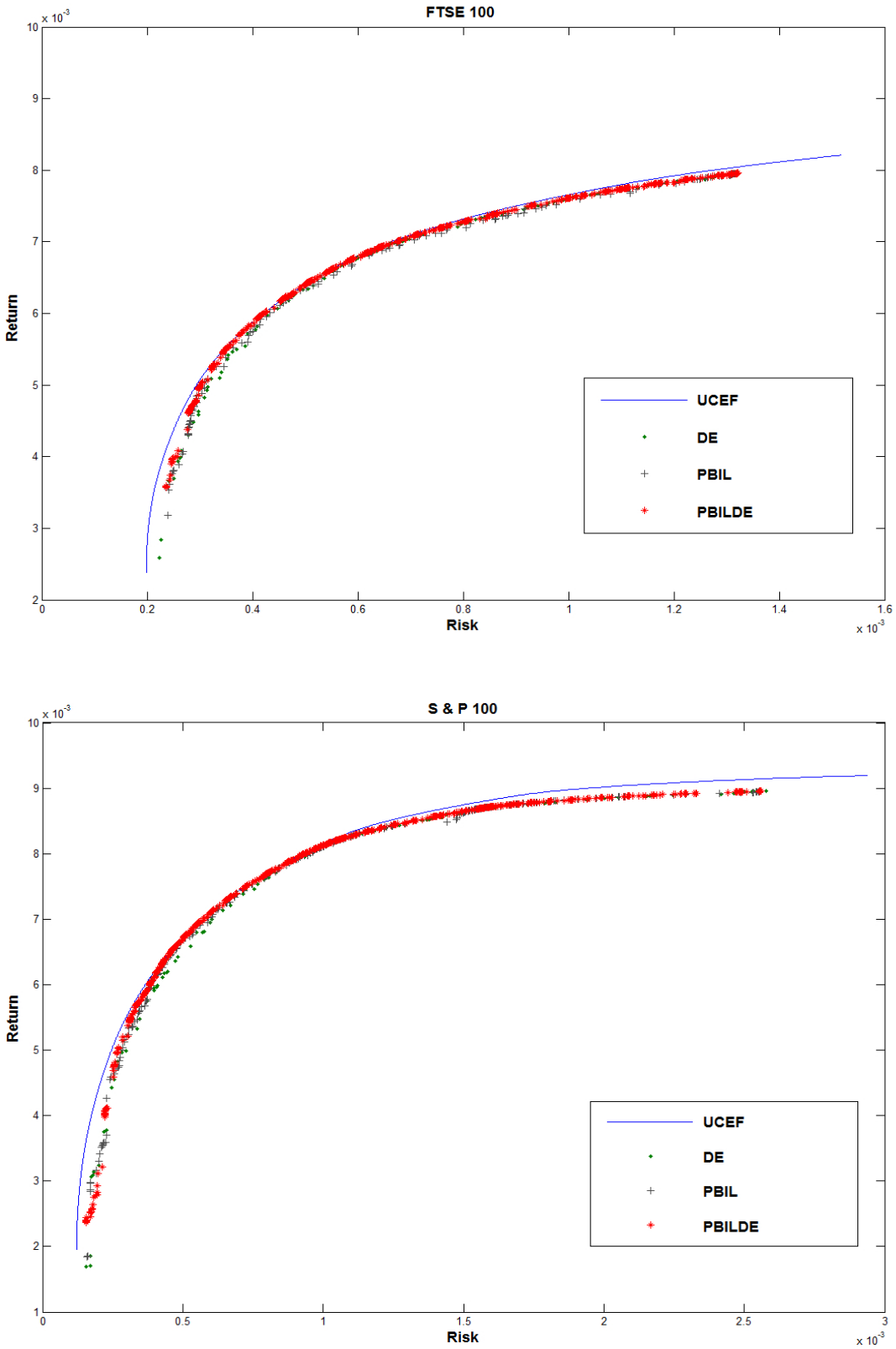


Figure 4.2: Comparison of heuristic efficient frontiers for constrained problem.

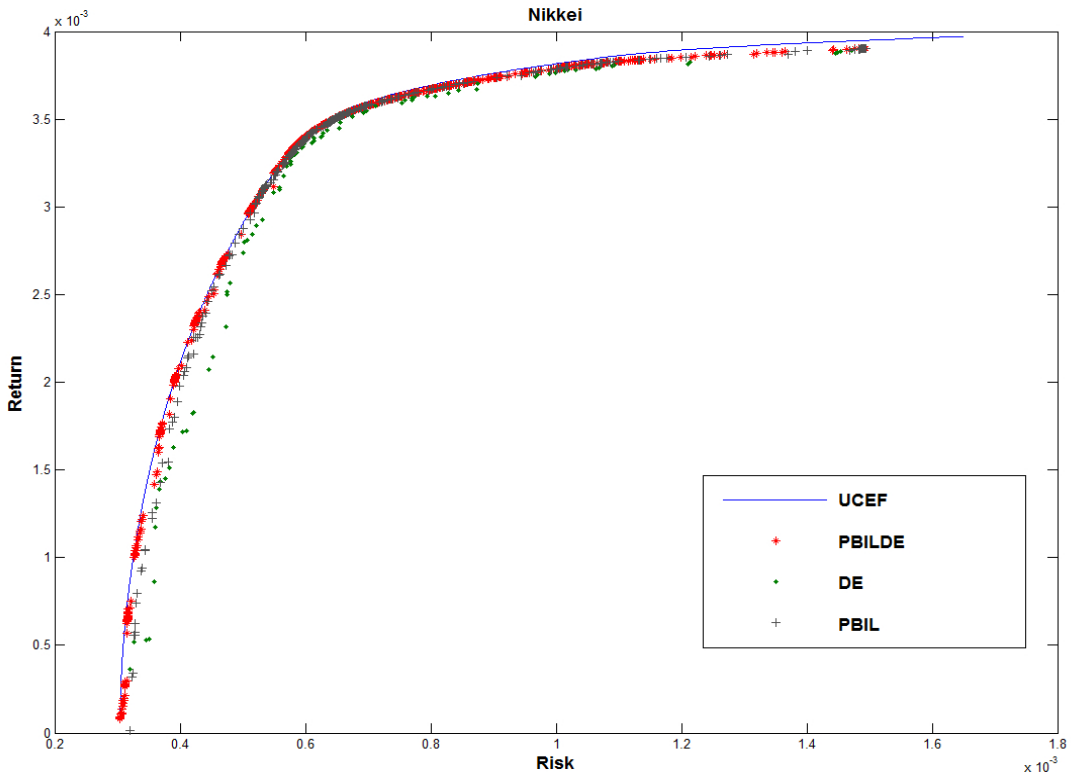


Figure 4.2: Comparison of heuristic efficient frontiers for constrained problem.

We also evaluated the performance of the algorithms by the average fitness of the efficient portfolios obtained throughout the evolution. The fitness of the algorithm in a certain generation is measured by the average mean percentage error deviation of the obtained efficient portfolios from the unconstrained efficient frontier (UCEF). The performance of the algorithms is provided in Figure 4.3. In all figures, the graphs represent the average of the mean percentage error in 20 runs. The results clearly demonstrate that our proposed algorithm PBILDE significantly outperforms DE and PBIL on all problems tested. Therefore, we could conclude that PBILDE is able to achieve a synergetic effect through hybridization of PBIL and DE.

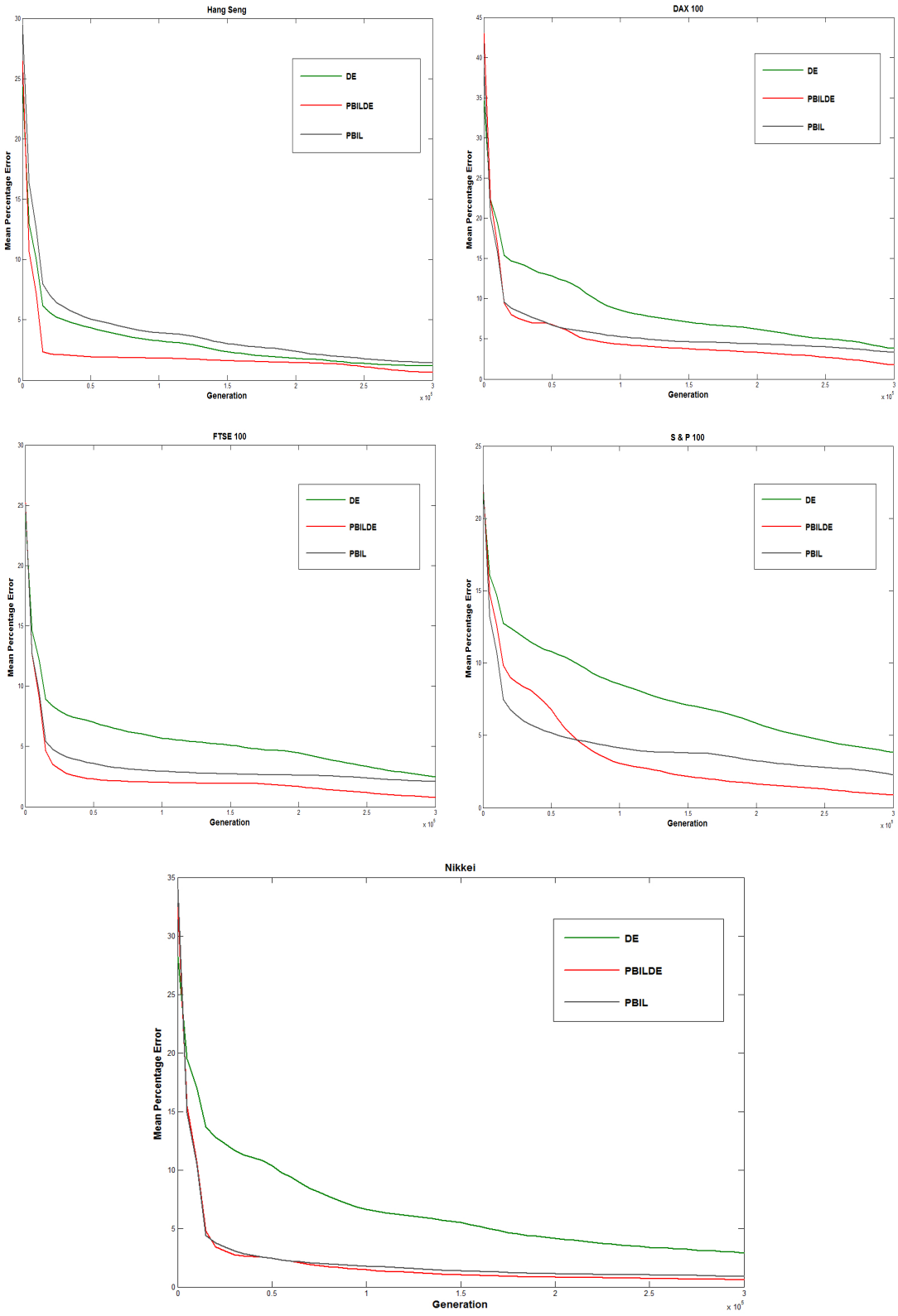


Figure 4.3: Mean performance of the algorithms for constrained problem.

Chang et al. (2000) presented three heuristic algorithms based on GA, SA and TS for the constrained problem and reported that GA performs better than SA and TS. Xu et al. (2010) also presented a hybrid algorithm (PBIL_CCPS) and reported that PBIL_CCPS performs better than GA and PSO. We therefore compare PBILDE with the GA proposed by Chang et al. (2000) and PBIL_CCPS presented by Xu et al. (2010) for the constrained problem. Both Chang et al. (2000) and Xu et al. (2010) adopted the *same* CCMV model described in Section 4.2. The results reported by their studies were obtained by using the same number of fitness evaluations and same set of λ values. The comparison results in Table 4.8 show that PBILDE outperforms GA and PBIL_CCPS in most instances.

			PBILDE		Chang-GA		Xu-PBIL_CCPS	
Index	$\mathcal{N}$		V	H	V	H	V	H
Hang Seng	31	MPE(%)	1.1431	**0.6196**	**1.0974**	0.9457	1.1026	0.8472
		MedPE(%)	1.2390	**0.4712**	**1.2181**	1.1819	1.2190	1.1013
		Number of EF points	**6367**		1317		1540	
DAX 100	85	MPE(%)	**2.4251**	**1.5433**	2.5424	1.9515	2.5163	2.0781
		MedPE(%)	2.5866	**1.0986**	**2.5466**	2.1262	2.5739	2.2783
		Number of EF points	**3378**		1270		1933	
FTSE 100	89	MPE(%)	**0.9706**	0.8234	1.1076	0.8784	0.9960	**0.7658**
		MedPE(%)	**1.0840**	0.5134	1.0841	0.5938	1.0841	**0.4132**
		Number of EF points	**2957**		1482		1638	
S&P 100	98	MPE(%)	**1.6386**	**1.3902**	1.9328	1.7157	2.2320	1.6340
		MedPE(%)	1.1692	**0.7303**	1.2244	1.1447	**1.1536**	0.8453
		Number of EF points	**4570**		1560		2177	
Nikkei	225	MPE(%)	**0.5972**	**0.3996**	0.7961	0.6431	1.0017	0.6451
		MedPE(%)	0.5896	**0.4619**	0.6133	0.6062	**0.5854**	0.5596
		Number of EF points	**4000**		1823		1468	
Average		avg MPE(%)	**1.3549**	**0.9552**	1.4953	1.2269	1.5697	1.1940
		avg MedPE(%)	1.3337	**0.6551**	1.3373	1.1306	**1.3232**	1.0395

Table 4.8: Comparison results of PBILDE against other existing algorithms (Chang et al., 2000; Xu et al., 2010) for the constrained problem.

Various models have been proposed in the literature to solve the constrained portfolio optimization problems, where different variable definitions, objective functions, heuristic techniques, benchmarks and evaluation criteria have been employed. Therefore, it is very difficult, if not impossible, to conduct a fair comparison on different modelling approaches. For completeness, we next provide the comparisons of our PBILDE against those of *different* approaches in Gaspero et al. (2011) and Woodside-Oriakhi et al. (2011) who use the OR-library instances with the same set of constraints.

Gaspero et al. (2011) presented a hybrid technique (SD+QP) which combines local search metaheuristics and the quadratic programming (QP) procedure. In their work, they also reimplemented the hybrid method based on a Hopfield neural network, originally proposed by Fernández and Gómez (2007), and calculated the mean percentage deviation in set H.

We compare PBILDE with this SD+QP approach by Gaspero et al. (2011) and the results are shown in Table 4.9. The comparison results show that PBILDE outperforms the SD+QP approach by Gaspero et al. (2011). As reported in Table 4.9, the neural network approach by Fernández and Gómez (2007) performs better than PBILDE in 3 out of 5 instances. However, PBILDE is better with regard to the overall average percentage error of all instances. It should be noted that Gaspero et al. (2011) adopted QP approach and results were obtained by 100 different return $\Re$ values (see QP model in Section 2.2) while results in this work were obtained from 50 λ values.

			PBILDE	Gaspero-SD+QP	Fernandez-NN
Index	$\mathcal{N}$		H	H	H
Hang Seng	31	MPE(%)	0.6196	0.7000	**0.3800**
DAX 100	85	MPE(%)	1.5433	2.9300	**1.1300**
FTSE 100	89	MPE(%)	**0.8234**	1.9700	1.2500
S&P 100	98	MPE(%)	**1.3902**	4.1000	2.8000
Nikkei	225	MPE(%)	0.3996	0.3000	**0.3600**
Average		MPE(%)	**0.9552**	2.000	1.1840

Table 4.9: Comparison results of PBILDE against Gaspero et al. (2011) and Fernández and Gómez (2007) for the constrained problem.

Recently, Woodside-Oriakhi et al. (2011) proposed a GA with subset optimization for the constrained problem. They adopted the QP approach and the constrained portfolio selection problem was reformulated by relaxing constraint (see Section 2.2, Eq. (2.2)), where the expected return may vary within 10% of the desired return range. The search of the algorithm is thus more flexible to explore a wider area of the search space of the relaxed problem. The same mechanism has been applied to develop a SA and TS. The weighted sum approach in this work approximates the constrained efficient frontier (CCEF) by accumulating the set of points which are unlikely to be evenly distributed along the return axis whereas Woodside-Oriakhi et al. (2011) approximates the CCEF by accumulating the set of efficient points which are evenly distributed among 50 return $\mathcal{R}$ values with each return in the pre-specified range.

The comparison results are shown in Table 4.10. The GA by Woodside-Oriakhi et al. (2011) outperforms in all instances except the Hang Seng dataset. PBILDE outperforms the SA in most instances and competitive to the TS by Woodside-Oriakhi et al. (2011). However, the maximum and minimum percentage error

results show that PBILDE results are stable compared to those of the three algorithms presented by Woodside-Oriakhi et al. (2011).

			PBILDE	Woodside-Oriakhi-GA	Woodside-Oriakhi-TS	Woodside-Oriakhi-SA
Index	N		H	H	H	H
Hang Seng	31	MPE(%)	**0.6196**	0.8501	0.8234	1.0589
		MedPE(%)	0.4712	0.5873	**0.3949**	0.5355
		Minimum	0.2816	**0.0036**	0.0068	0.0349
		Maximum	**0.6768**	2.9034	4.6096	4.6397
DAX 100	85	MPE(%)	1.5433	0.7740	**0.7190**	1.0267
		MedPE(%)	1.0986	**0.2400**	0.4298	0.8682
		Minimum	0.7537	**0.0000**	0.0149	0.0278
		Maximum	**1.6804**	4.6811	2.7770	4.4123
FTSE 100	89	MPE(%)	0.8234	**0.1620**	0.3930	0.8952
		MedPE(%)	0.5134	**0.0820**	0.2061	0.3944
		Minimum	0.4359	**0.0000**	0.0019	0.0230
		Maximum	0.8695	**0.7210**	3.4570	10.2029
S&P 100	98	MPE(%)	1.3902	**0.2922**	1.0358	3.0952
		MedPE(%)	0.7303	**0.1809**	1.0248	2.1064
		Minimum	0.4816	**0.0007**	0.0407	0.8658
		Maximum	**1.5726**	1.6295	3.0061	8.6652
Nikkei	225	MPE(%)	0.3996	**0.3353**	0.7838	1.1193
		MedPE(%)	0.4619	**0.3040**	0.6526	0.6877
		Minimum	0.3739	0.0180	**0.0085**	0.0113
		Maximum	**0.4965**	1.0557	2.6082	3.9678
Average		MPE(%)	0.9552	**0.4827**	0.7510	1.4391
		MedPE(%)	0.6550	**0.2788**	0.5416	0.9184
		Minimum	0.4653	**0.0045**	0.0146	0.1926
		Maximum	**1.0591**	2.1981	3.2916	6.3776

Table 4.10: Comparison results of our Hybrid Algorithm(PBILDE) against Woodside-Oriakhi et al (Woodside-Oriakhi et al., 2011) for the constrained problem.

4.6 Summary and Discussion

This chapter presents a new efficient and effective hybrid algorithm (PBILDE) to solve the mean variance portfolio selection problem with cardinality and quantity constraints. The proposed algorithm hybridizes DE and PBIL to explore and exploit the complex and constrained search space of the problem concerned. It also adopts a partially guided mutation and an elitist strategy to promote the efficient convergence of the search. The partially guided mutation is introduced not only to exploit the global information about the search space from the probability vector but also to exploit the information from the best solution to guide the search. The proposed elitist strategy makes use of a set of global best solutions to inject into the current population as a replacement with its worst members when the current best solution is worse than the global best solution. This strategy is introduced in order to reduce the chances of being stuck in the local optima.

Computational results justify the effectiveness of the elitism and partially guided mutation in PBILDE. For the unconstrained problems, PBILDE outperforms in almost all instances compared against DE and PBIL with similar or higher computational expenses. In most problem instances, it also outperforms other existing approaches in the literature for the unconstrained problem. The comparison results against the PBIL, DE, as well as Chang-GA (Chang et al., 2000) and Xu-PBIL_CCPS (Xu et al., 2010) in the literature also show that the proposed hybrid algorithm is highly competitive in most cases. Results also show that PBILDE is able to achieve a synergetic effect through hybridization of PBIL and DE.

In this work, weighted sum approach is utilized to transform the bi-objective portfolio optimization problem into a scalar optimization problem. Despite its simplicity, there are a few drawbacks with this approach:

- Despite its insight of the relative importance of objective, it is difficult to identify the appropriate weights needed for each objective in order to generate solutions uniformly spread on the efficient frontier. Applying a uniform set of weighting parameters does not produce solutions (in the objective space) evenly spread on the Pareto front. Moreover, small pertur-

bations of weights can occasionally lead to quite different solutions (Konak et al., 2006). Therefore, it is hard to compare with other studies unless the same set of weights are set for the computational analysis.

- It cannot find solutions which reside in non-convex regions of the Pareto front (Das and Dennis, 1997; Kim and de Weck, 2005). Chang et al. (2000) and Jobst et al. (2001) showed that the efficient frontier become discontinuous in the presence of cardinality constraints.

- It requires repeated runs of the algorithm in order to find the efficient frontier and hence it is time consuming (see Section 3.2.2) (Anagnostopoulos and Mamanis, 2011b; Marler and Arora, 2010).

On the other hand, studies on the multi-objective evolutionary algorithms (MOEAs) have shown that MOEAs can yield multiple Pareto optimal solutions in a single run. In addition, they require very little knowledge of the problem being solved. Therefore, multi-objective approaches for the constrained portfolio optimization problems will be studied in the subsequent chapters in order to alleviate the difficulties faced by the parameter-oriented scaling dependent weighted sum approach.

Chapter 5

Multi-objective Scatter Search for Portfolio Optimization

> "Risk comes from not knowing what you're doing. "
>
> Warren Buffett

5.1 Introduction

As noted in the previous chapter, single objective optimization approaches can provide a single efficient solution in each run. As a result, they are computationally expensive to investigate the solution search space and require many repeated runs to compute the efficient frontier. Moreover, they do not consider a good distribution of the obtained solutions nor find Pareto optimal solutions in non-convex regions. In contrast, multi-objective optimization approaches operate on a number of solutions (population) and are capable to find several efficient solutions in a single run. In addition, they are less susceptible to the shape or continuity of the Pareto front (Das and Dennis, 1997). Anagnostopoulos and Mamanis (2011b) also showed that all five MOEAs tested in their work outperformed a single objective evolutionary algorithm (SOEA) in all tested problem instances. Moreover, the results of MOEAs need fewer number of solution generations and less computational time than SOEA. These studies have helped us to

understand the limitations of the SOEA and paved ways to develop new and improved multi-objective evolutionary algorithms for portfolio optimization problems.

In this chapter, we present a hybrid multi-objective population-based evolutionary algorithm based on Scatter Search template to solve the portfolio optimization problem with three practical trading constraints, namely cardinality, quantity and pre-assignment. In the literature, not many studies have been conducted by taking into account of pre-assignment constraint for portfolio optimization problem. Although Gaspero et al. (2011) considered cardinality, quantity and pre-assignment constraints in their hybrid model, the experiments were not performed by considering all three constraints together. To the best of our knowledge, there is no comparative study of multi-objective evolutionary algorithm based on scatter search template for the portfolio optimization problem with cardinality, quantity and pre-assignment constraints. This study is intended to fill in this gap.

5.2 Problem Model

The basic MV model is extended with three practical constraints, cardinality, quantity and pre-assignment as follows:

$$min \qquad f_1 = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} w_i w_j \sigma_{ij} \tag{5.1}$$

$$max \qquad f_2 = \sum_{i=1}^{\mathcal{N}} w_i \mu_i \tag{5.2}$$

$$subject\ to \quad \sum_{i=1}^{\mathcal{N}} w_i = 1 \tag{5.3}$$

$$\sum_{i=1}^{\mathcal{N}} s_i \ \leq \ \mathcal{K}, \tag{5.4}$$

$$\epsilon_i s_i \leq w_i \leq \delta_i s_i, \quad i = 1, ..., \mathcal{N}, \tag{5.5}$$

$$s_i \geq z_i, \quad i = 1, ..., \mathcal{N}, \tag{5.6}$$

$$s_i \in \{0, 1\}, \quad i = 1, ..., \mathcal{N}, \tag{5.7}$$

$$z_i \in \{0, 1\}, \quad i = 1, ..., \mathcal{N}, \tag{5.8}$$

where f_1 and f_2 are risk and return objectives respectively (see Section 2.2.2), $\mathcal{K}$ is the maximum number of assets which can be invested in a portfolio, s_i denotes whether asset i is invested or not, z_i denotes a binary value such that z_i is set to one if asset i is included in the pre-assigned set in the portfolio. If s_i equals one, asset i is chosen to be invested and the proportion of capital w_i lies in $[\epsilon_i, \delta_i]$, where $0 \leq \epsilon_i \leq \delta_i \leq 1$. Otherwise, asset i is not invested and w_i equals zero. The objective is to find the efficient portfolios among $\mathcal{N}$ assets that can *simultaneously* satisfy the two conflicting objectives, i.e., minimize risk f_1 while maximizing the profit f_2.

5.3 Related Work

In the literature, not many studies have been performed by taking into account of pre-assignment constraints. Chang et al. (2000) and Di Tollo and Roli (2008) described the pre-assignment constraints but they are not considered in the computational experiments of their work.

Gaspero et al. (2011) considered the pre-assignment, cardinality and quantity constraints. Their work considered the cardinality constraint with minimum and maximum limits on the number of assets in a portfolio. The authors presented a hybrid technique that combines a local search with quadratic programming procedure. Two groups of experiments for the portfolio selection problem (PSP)

were performed in their work: PSP with cardinality and quantity constraints and PSP with pre-assignment and quantity constraints. The experiments were not performed with all three constraints. The experimental results showed that the hybrid technique achieved comparable or superior results compared with commercial software tools (CPLEX and MOSEK).

In the literature, many researchers have investigated a variety of techniques to solve the portfolio optimization problem with cardinality (with inequality) and quantity constraints. There are several studies employing exact approaches (Bertsimas and Shioda, 2009; Li et al., 2006; Shaw et al., 2008; Vielma et al., 2008). In the remaining part of this section, we review the studies which adopted heuristic approaches and considered the cardinality constraint (with inequality) and/or quantity constraints.

Single objective approaches

Crama and Schyns (2003) presented a simulated annealing (SA) approach for the complex portfolio optimization problem with cardinality, quantity, turnover and trading constraints. Their work considered inequality cardinality constraints by limiting the maximum number of assets allowed in a portfolio. Their work adopted different constraint handling approaches (both repair and penalty function) based on the types of constraints. Computational experiments were performed on a realistic problem instance involving 151 assets on each class of considered constraints separately. Experimental results compared against simplex methods showed promising. Maringer and Kellerer (2003) also presented a hybrid approach incorporating SA with evolutionary strategies. They considered the cardinality constraints and computational analyses were presented for two test instances involving 30 and 96 assets.

Ruiz-Torrubiano and Suarez (2010) presented hybrid algorithms that combine EAs and QP with specially devised pruning heuristics. In their approach, SA, GA and EDAs were employed to find the promising subset of assets to be included in a portfolio and QP was utilized to find the optimal weights. Their work consid-

ered (inequality) cardinality and quantity constraints and OR-library instances are used for experimental analysis. Results showed that without using pruning heuristics, EDAs (Larrañaga and Lozano, 2002) do not perform well on large problem instances and pruning heuristics generally improved all considered algorithms both in terms of the computation time and solution quality.

Schaerf (2002) compared hill climbing (HC), tabu search (TS) and simulated annealing (SA) algorithms for the portfolio optimization problem with cardinality and quantity constraints. A variety of moves for the invested proportion of assets are presented. These moves ensure that the budget constraint is satisfied. Performances of the algorithms are tested by employing the OR-library datasets (Beasley, 1999) and results showed that TS outperformed HC and SA. Busetti Busetti (2005) also investigated GA and a hybrid technique that combines SS and TS to solve the portfolio optimization problem with cardinality, bounding and transaction cost constraints. The results showed that GA outperformed the hybrid approach.

Multi-objective approaches

Fieldsend et al. (2004) and Anagnostopoulos and Mamanis (2010) presented a tri-objective view of the portfolio optimization problem: reward, risk and the number of assets in a portfolio. Fieldsend et al. (2004) considered the basic model and applied a MOEA to find a discrete approximation of the efficient surface in a single run. Anagnostopoulos and Mamanis (2010) considered quantity and class constraints and applied three MOEAs, namely NSGA-II, SPEA2 and PESA to find a good approximation of the efficient surface.

Anagnostopoulos and Mamanis (2011b) presented a computational comparison of the five MOEAs for the portfolio optimization problems with cardinality and quantity constraints. The results showed a clear superiority of SPEA2 in most problem instances. They also performed a comparison of MOEAs with a variant of SOEA and results showed that all five MOEAs are superior than SOEA both in terms of solution quality and computational time.

Branke et al. (2009) presented a hybrid envelope-based approach by incorporating Markowitz's critical line algorithm (CLA) with NSGA-II. NSGA-II was utilized to define suitable $convex$ subsets of the search space and CLA was then applied to each subset thereby generating an envelope. All generated envelopes by CLA are then combined to compute the constrained efficient frontier. Their work considered cardinality constraints based on German investment law. They showed that the proposed envelope-based NSGA-II finds better frontiers in a shorter time than point-based NSGA-II.

Chiam et al. (2008) presented a MOEA with an order-based representation. Their work considered cardinality and quantity constraints. The cardinality constraint is not only relaxed with the minimum and maximum number of assets that a portfolio can hold but also considered as a soft constraint (i.e., population may be composed of infeasible solutions). Experiments are performed on the OR-library dataset instances (Beasley, 1999) and the results showed that MOEA with an order-based representation could find solutions close to the unconstrained efficient frontier.

Deb et al. (2011) suggested a customized hybrid NSGA-II integrated with a clustering and local search procedure. They considered cardinality and quantity constraints. A repair mechanism was adopted to ensure that all solutions in the evolving process are feasible. The results showed that it is competitive to QP solutions.

Liagkouras and Metaxiotis (2014) presented a new probe guided mutation operator for efficient exploration of the search space. Their work considered the cardinality constraints limiting the minimum and maximum number of assets in a portfolio and the quantity constraints. Their proposed probe guided mutation operator was incorporated into NSGA-II and SPEA2 and experimental results using the OR-library datasets confirmed the efficient contribution of the probe guided mutation operator.

Skolpadungket et al. (2007) applied various techniques of MOEAs to solve port-

folio optimization with cardinality, floor and round lot constraints. They integrated the Vector Evaluated Genetic Algorithm (VEGA) with fuzzy logic to investigate the performances of the solutions. Results indicated that solution quality improved in terms of closeness to the true Pareto front but not in terms of distribution. Experiments were performed on the smallest OR-library dataset $D1$ and performance metrics showed that SPEA2 outperformed the others.

Decomposition approach

Zhang et al. (2010) presented a multi-objective evolutionary algorithm based on decomposition (MOEA/D) with NBI-style Tchebycheff approach. Their work considered inequality cardinality constraints which limit the maximum number of assets in a portfolio and three variants of transaction cost constraints. Experiments are performed on the eight instances involving up to 150 assets and results showed that MOEA/D outperformed NSGA-II in unconstrained cases and showed promising in constrained cases.

5.4 Multi-objective Scatter Search with External Archive

In this work, we present a hybrid multi-objective scatter search with an external archive, MOSSwA, to solve the portfolio optimization model described in Section 5.2. This work has been published at ECTA2013 (Lwin et al., 2013).

MOSSwA adapts the basic scatter search template to multi-objective optimization by incorporating the concepts of Pareto dominance, crowding distance and elitism. It follows the basic structure of the scatter search (see Section 3.3.1.3) and defines the reference set solutions based on Pareto dominance and crowding distance measures. New Subset generation and combination methods are proposed to generate efficient and diverse portfolios. Hill climbing operation is integrated to search for improved portfolios. The detailed procedure of the proposed algorithm is described in Algorithm 5.1.

Algorithm 5.1: Procedure of MOSSwA.

Input: $\mathcal{A}_{size}$: the size of the archive A,
$\mathcal{NP}$: the number of individuals in the population P,
$\mathcal{B}$: the number of solutions in reference sets ($ref1$ and $ref2$);
Output: A : archived non-dominated set of solutions;

$P = \emptyset;\ A = \emptyset;\ ref1 = \emptyset;\ ref2 = \emptyset;$
for $j = 1$ *to* $\mathcal{NP}$ **do** `// Initialization (see Section 5.4.2)`
  $p_j \leftarrow$ generate a random individual including pre-assigned asset(s);
  if *constraints are violated* **then**
    $p_j \leftarrow$ repair by constraint handling method (see Section 5.4.8);
  $P \leftarrow P \cup p_j;$
repeat
  $A \leftarrow$ update with $\mathcal{A}_{size}$ non-dominated portfolios from $(A \cup P)$;
  `// Update reference sets (see Section 5.4.7)`
  $ref1 \leftarrow$ select $\mathcal{B}$ non-dominated portfolios from $(A \cup ref1)$;
  $ref2 \leftarrow$ select $\mathcal{B}$ non-redundant and least crowded portfolios
    from $(A \setminus ref1) \cup P$;
  $P = \emptyset;$
  for $j = 1$ *to* $\mathcal{NP}$ **do**
    `// Generate subset (see Section 5.4.3).`
    $sub \leftarrow$ randomly select $\mathcal{S}$ portfolios from $(ref1 \cup ref2)$;
    `// Combine solution (see Section 5.4.4)`
    $p_j \leftarrow$ generate a new portfolio by solution combination method;
    if *constraints are violated* **then**
      $p_j \leftarrow$ repair by constraint handling method;
    $p_j^{'} \leftarrow$ apply local search to p_j (see Section 5.4.5);
    if $p_j \preceq p_j^{'}$ **then**
      $P \leftarrow P \cup p_j;$
    else if $p_j^{'} \preceq p_j$ **then**
      $P \leftarrow P \cup p_j^{'};$
    else
      $P \leftarrow P \cup p_j^{'} \cup p_j;$
until *(certain number of generations)*;

5.4.1 Solution Representation

Two vectors of size $\mathcal{N}$ are used to define a portfolio p: a binary vector $s_i, i = 1, \ldots, \mathcal{N}$ denoting whether asset i is included in the portfolio, and a real-value vector $w_i, i = 1, \ldots, \mathcal{N}$ representing the proportions of the capital invested in the assets.

5.4.2 Initialization

Each portfolio p of an initial population $P = \{p_1, p_2, \ldots, p_{\mathcal{NP}}\}$ is generated by randomly selecting the maximum $\mathcal{K}$ different assets (including the pre-assign assets) and allocating the weights for those selected assets. In this work, we consider all three constraints as *hard* ones which need to be satisfied at all times. If the generated portfolio violates the budget and/or quantity constraints (see Eqs. 5.3 and 5.5), such solution is corrected by the constraint handling techniques as previously described in Section 4.4.7. As a result, all of the generated solutions in the trial population are feasible.

5.4.3 Subset Generation Method

This method selects a subset of solutions from the reference sets to create a subset $sub = \{p_1, \ldots, p_{\mathcal{S}}\}$ which is later used by the combination method (see Section 5.4.4). Our approach generates the subset sub of size $\mathcal{S}$ by randomly employing one of the three different variants as follows:

- randomly select $\mathcal{S}$ solutions from the reference set $ref1 = \{p_1, \ldots, p_{\mathcal{B}}\}$:

  $$sub = \{p_i \in ref1 \mid i = randint[1, \mathcal{B}]\},$$

- randomly select $\mathcal{S}$ solutions from the reference set $ref2 = \{p_1, \ldots, p_{\mathcal{B}}\}$:

  $$sub = \{p_i \in ref2 \mid i = randint[1, \mathcal{B}]\},$$

- randomly select arbitrary r' solutions from the reference set $ref1$ and $\mathcal{S} - r'$ solutions from the reference set $ref2$ where $r' \leq \mathcal{S}$ and $\mathcal{S} \geq 3$.

5.4.4 Solution Combination

The combination method generates individuals by sampling the candidate solutions towards the areas of the search space which are likely to be of high fitness by exploiting the information present in the population and the archive A. This method uses the generated subset sub (see Section 5.4.3) and combines solutions from sub to generate one or more trial solutions.

Three portfolios $p1, p2$ and $p3$ from the sub are randomly selected to use in the combination process. The assets selected in these three portfolios are observed and analysed. A set q is constructed by composing with all securities which are selected by at least two out of three portfolios. A new portfolio with n assets is constructed by selecting pre-assigned assets first. The remaining assets $n- \mid z \mid$ are then randomly selected from the set q where $n \leq \mathcal{K}$. The proportions of those selected assets are assigned as follows:

$$w_i = w3_i + rand(0, 1] \times (w1 - w2)$$

If there are less assets in set q than $n- \mid z \mid$, the remaining $\bar{a}$ assets of the new portfolio are selected by one of the following methods:

- select $\bar{a}$ assets with the highest expected return values
- select $\bar{a}$ assets with the least standard deviation values
- select $\bar{a}$ assets with the lowest correlation values to those in the selected set

The weights of those $\bar{a}$ assets are randomly assigned ($w_i := rand(\epsilon_i, \delta_i)$). It is noted that the combination mechanisms construct solutions that may violate the budget and/or quantity constraints. Thus, the repair mechanism (see Section 5.4.8) is applied if the newly generated solution violates the budget and/or quantity constraints.

5.4.5 Improvement Method

This method adopts the 'local exploitation' by extracting the most important information possible at a local level and aims to move towards the local optimum. The solutions generated by the combination method (see Section 5.4.4) are improved by a local search technique. A simple hill climbing (HC) operation is employed by randomly altering a proportion of an asset or a selection of the asset in the portfolio until no improved solutions can be found or for a pre-specified number of moves.

5.4.6 Maintaining the Archive

The external archive A is used to reserve the well-spread non-dominated solutions encountered during the search. In each generation, the archive A is updated with the non-dominated solutions from the trial population. When it has reached its maximum capacity $\mathcal{A}_{size}$, the most crowded non-dominated members are identified and discarded.

5.4.7 Updating Reference Set

Once the archive A has been updated with non-dominated solutions from the trial population, the reference sets ($ref1$ and $ref2$) are updated by the improved and diverse solutions. The reference set $ref1$ is updated by the $\mathcal{B}$ best non-dominated solutions obtained from the archive A. The reference set $ref2$ is updated by the $\mathcal{B}$ non-redundant and least crowded portfolios from the remaining set of individuals in the archive A and from the current population P.

5.4.8 Constraint Handling

During the population sampling, each constructed individual must be repaired if the representative portfolio does not satisfy the constraints of the portfolio selection problem. As described in Section 5.4.4, the solution combination operation ensures that the pre-assignment and cardinality constraints are satisfied. However, the generated solution may violate the budget and/or quantity constraints.

The same constraint handling mechanism described in Section 4.4.7 has been applied to repair those solutions which violate the budget and/or quantity constraints.

5.5 Experimental Results

In this work, we compare the performance of the proposed MOSSwA with three well-known MOEAs, namely NSGA-II (Deb et al., 2002), SPEA2 (Zitzler et al., 2001) and PESA-II (Corne et al., 2001). The detailed descriptions of the three algorithms are provided in Section 6.5.2. All algorithms considered in this study are coded in C# and run on a core2duo with a 2.79GHz processor and 2GB RAM. Twenty independent runs are performed for all experiments and the same random seed is assigned to each set of the instance so that all algorithms start with the same initial population. In order to ensure a fair comparison, we use the same population size $\mathcal{NP}$ and archive size $\mathcal{A}_{size}$ (if applicable) for all the algorithms tested in this work. In addition, all the algorithms are run for the same stopping criteria (i.e. the same number of evaluations) to generate the efficient frontiers.

Five datasets (D1 $-$ D5) from OR-library (Beasley, 1990, 1999) (see Section 2.5) are used to evaluate the performance of the algorithms. Before the experiments were performed, parameters were tuned for all algorithms using the smallest problem instance, i.e. Hang Seng (D1). The parameter values of the tested algorithms are provided in Table 5.1. For constraint parameter values, we use $\mathcal{K} = 10$, $\epsilon_i = 0.01$, $\delta_i = 1$ $(i = i, ..., \mathcal{N})$ and $z = \{30\}$.

Parameters	MOSSwA	NSGA-II	SPEA2	PESA-II
Number of Population ($\mathcal{NP}$)	100	100	100	100
Number of Generation	4,000	4,000	4,000	4,000
Crossover Probability ($\mathcal{CR}$)	-	0.9	0.9	0.9
Crossover Distribution Index	-	20	20	20
Mutation Probability	-	$1/\mathcal{N}$	$1/\mathcal{N}$	$1/\mathcal{N}$
Mutation Distribution Index	-	20	20	20
Tournament Round	-	-	1	-
Number of Bisection	-	-	-	5
Reference set size ($\mathcal{B}$)	15	-	-	-
Subset sub size ($\mathcal{S}$)	3	-	-	-
Number of termination stage in HC	1	-	-	-
Archive Size ($\mathcal{A}_{size}$)	100	-	100	100

Table 5.1: Parameter setting of considered algorithms.

Three performance metrics, GD, IGD and HV, are used to evaluate the quality of the solutions achieved by considered algorithms (see Section 3.4). The experimental results of GD, IGD and Δ of the four MOEAs performed on the five datasets (D1 $-$ D5) are shown in Figures [5.1 - 5.5]. The results show that MOSSwA outperforms SPEA2, NSGA-II and PESA-II in all five problem instances in terms of GD, IGD and Δ metrics. SPEA2 is the second best algorithm out of the four algorithm tested. For the small problem instances (D1 $-$ D4), SPEA2 performs better than NSGA-II and PESA-II with higher computational cost. For Nikkei instance (D5), NSGA-II performs better than SPEA2 and PESA-II. In term of diversity measure (Δ), PESA-II is not able to achieve a good spread of efficient solutions for all five problem instances.

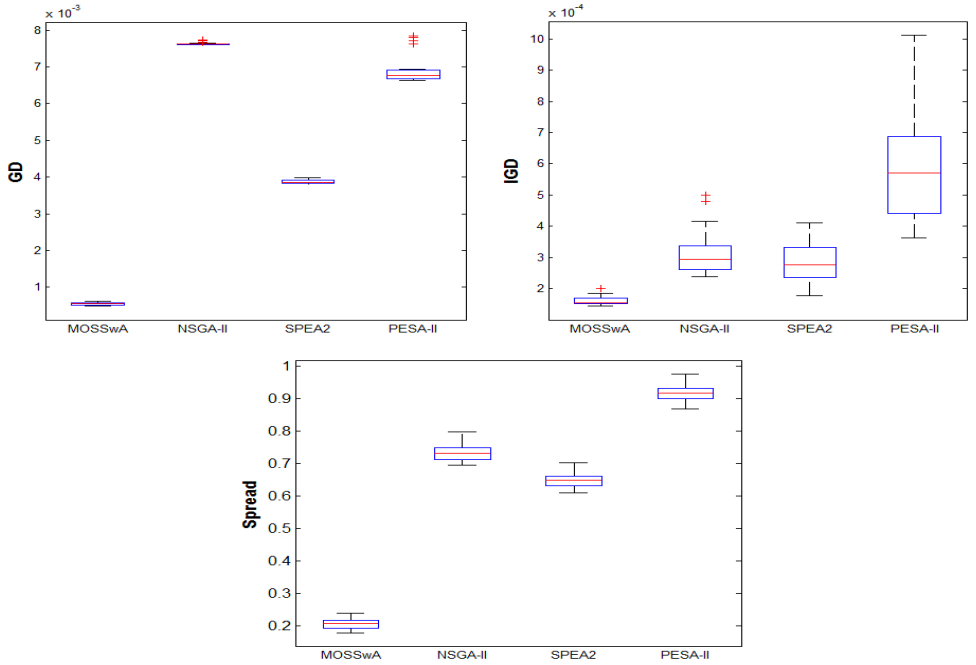


Figure 5.1: Performance comparisons of the algorithms in terms of GD, IGD and Spread (Δ) metrics for Hang Seng.

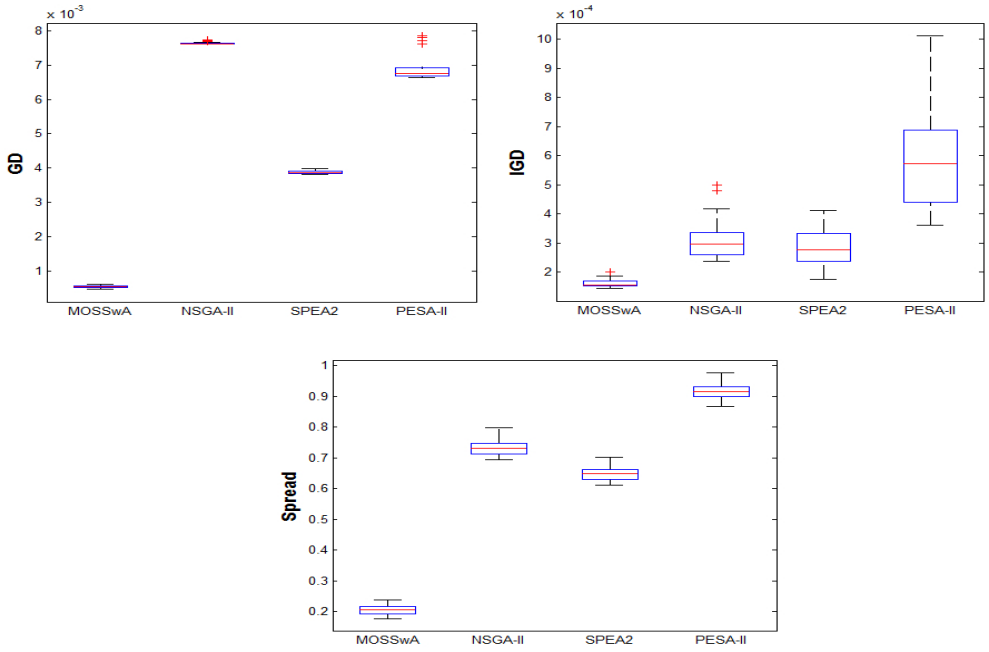


Figure 5.2: Performance comparisons of the algorithms in terms of GD, IGD and Spread (Δ) metrics for DAX 100.

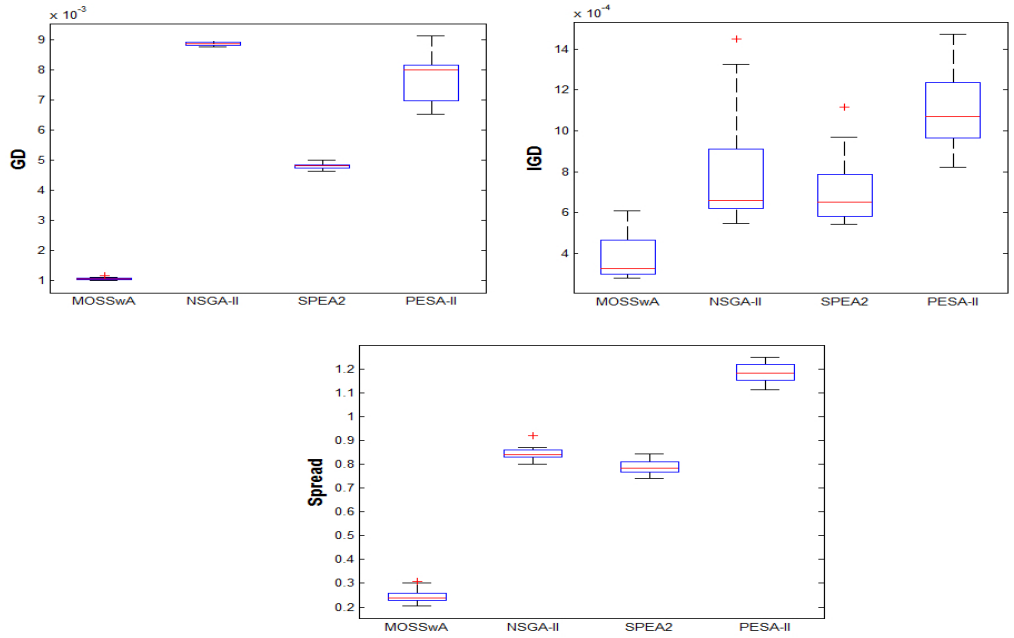


Figure 5.3: Performance comparisons of the algorithms in terms of GD, IGD and Spread (Δ) metrics for FTSE 100.

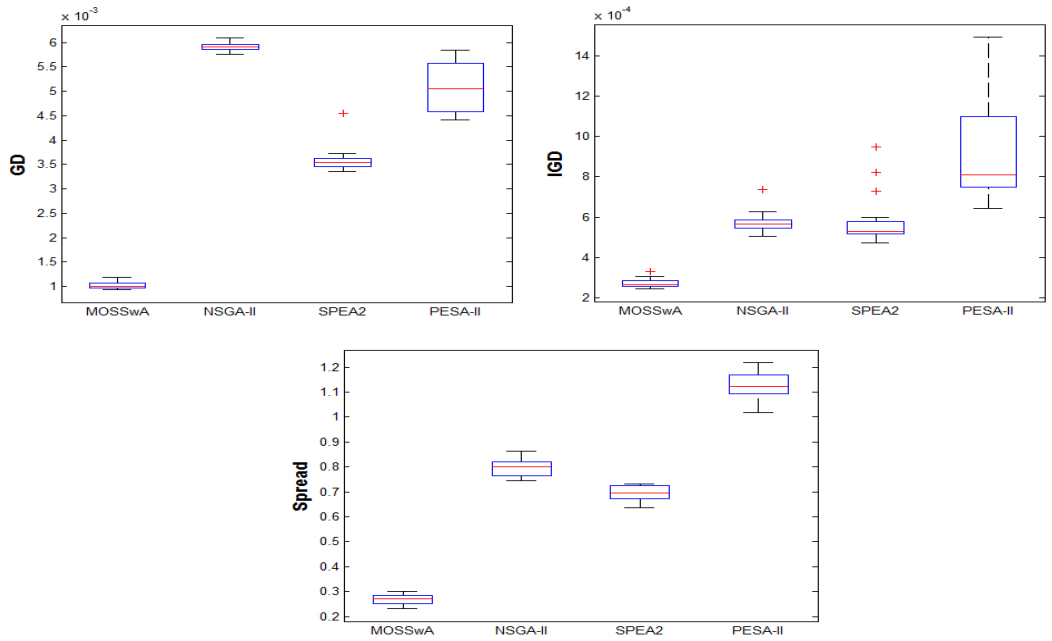


Figure 5.4: Performance comparisons of the algorithms in terms of GD, IGD and Spread (Δ) metrics for S&P 100.

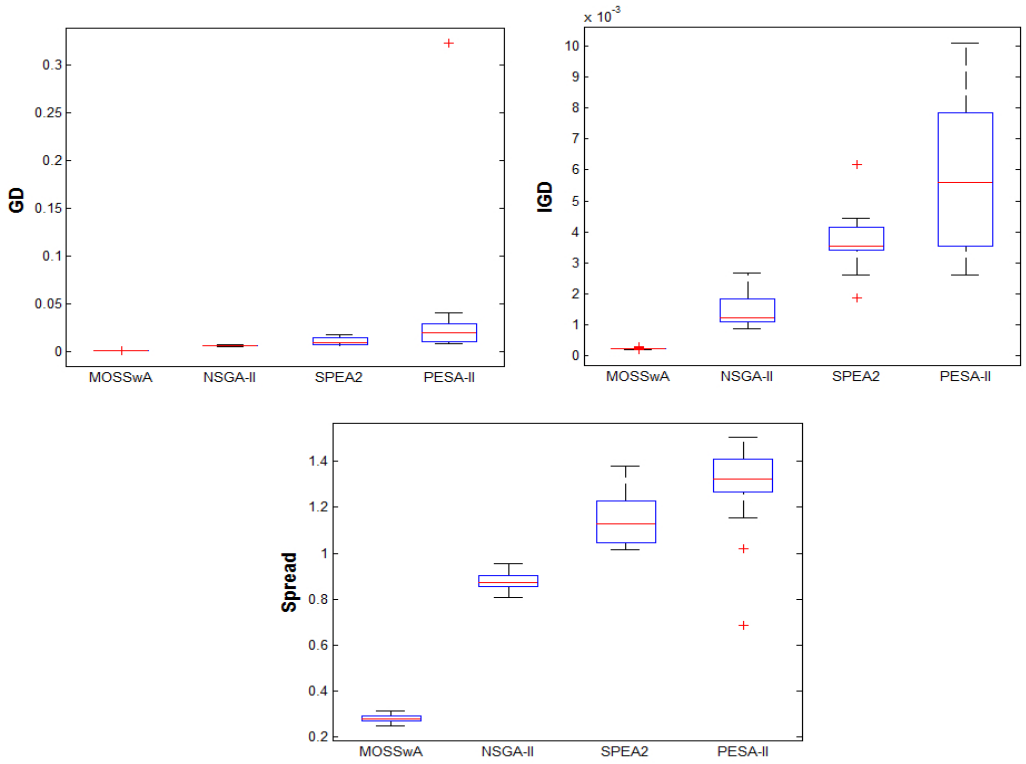


Figure 5.5: Performance comparisons of the algorithms in terms of GD, IGD and Spread (Δ) metrics for Nikkei.

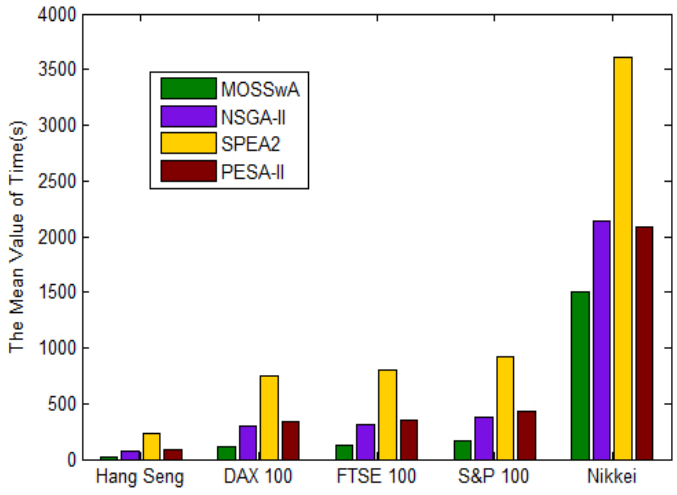


Figure 5.6: Running time of the algorithms for the constrained portfolio optimization problem.

The running times of the tested algorithms are shown in Figure 5.6. The results show that our proposed algorithm MOSSwA is not only superior in performance measures but also is efficient in computational time compared with NSGA-II, SPEA2 and PESA2 in all five datasets.

For illustrative purpose, the obtained efficient frontiers of the tested algorithms along with the unconstrained efficient frontier (UCEF) for five problem instances are provided in Figure 5.7. The results provided in Figure 5.7 are the obtained efficient frontiers from a single run.

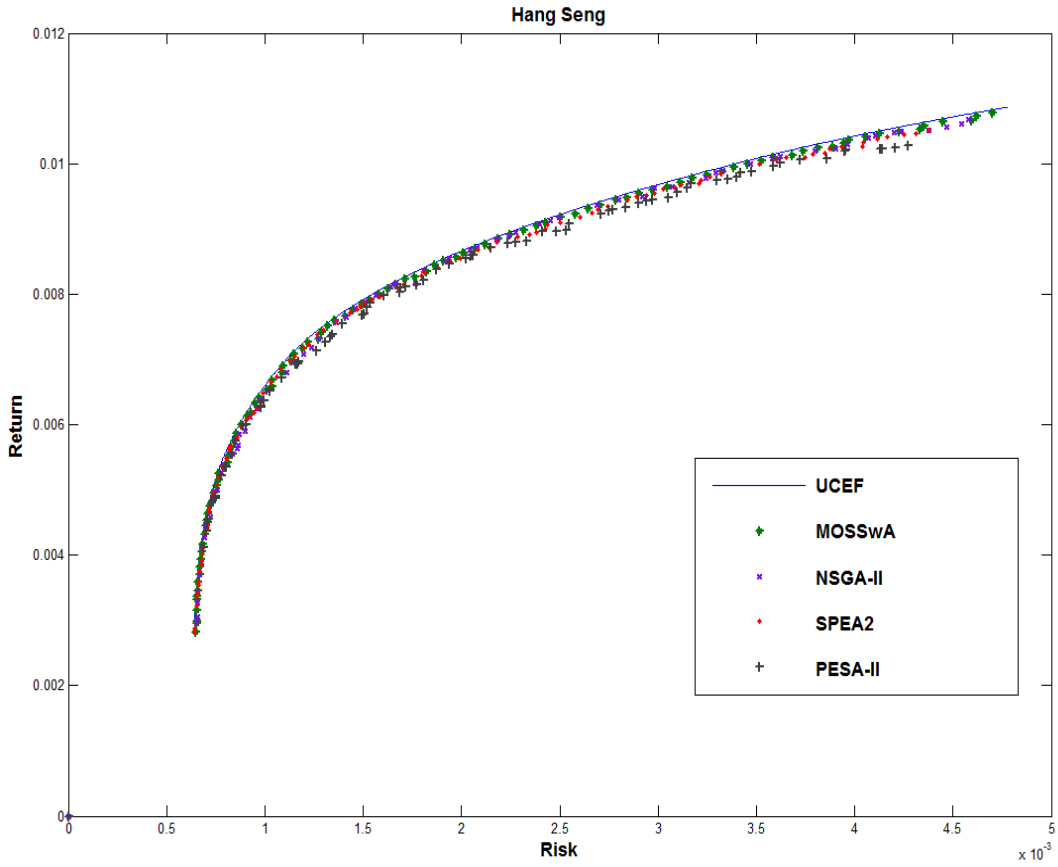


Figure 5.7: Comparison of obtained Efficient Frontier of all the algorithms for constrained portfolio optimization problem.

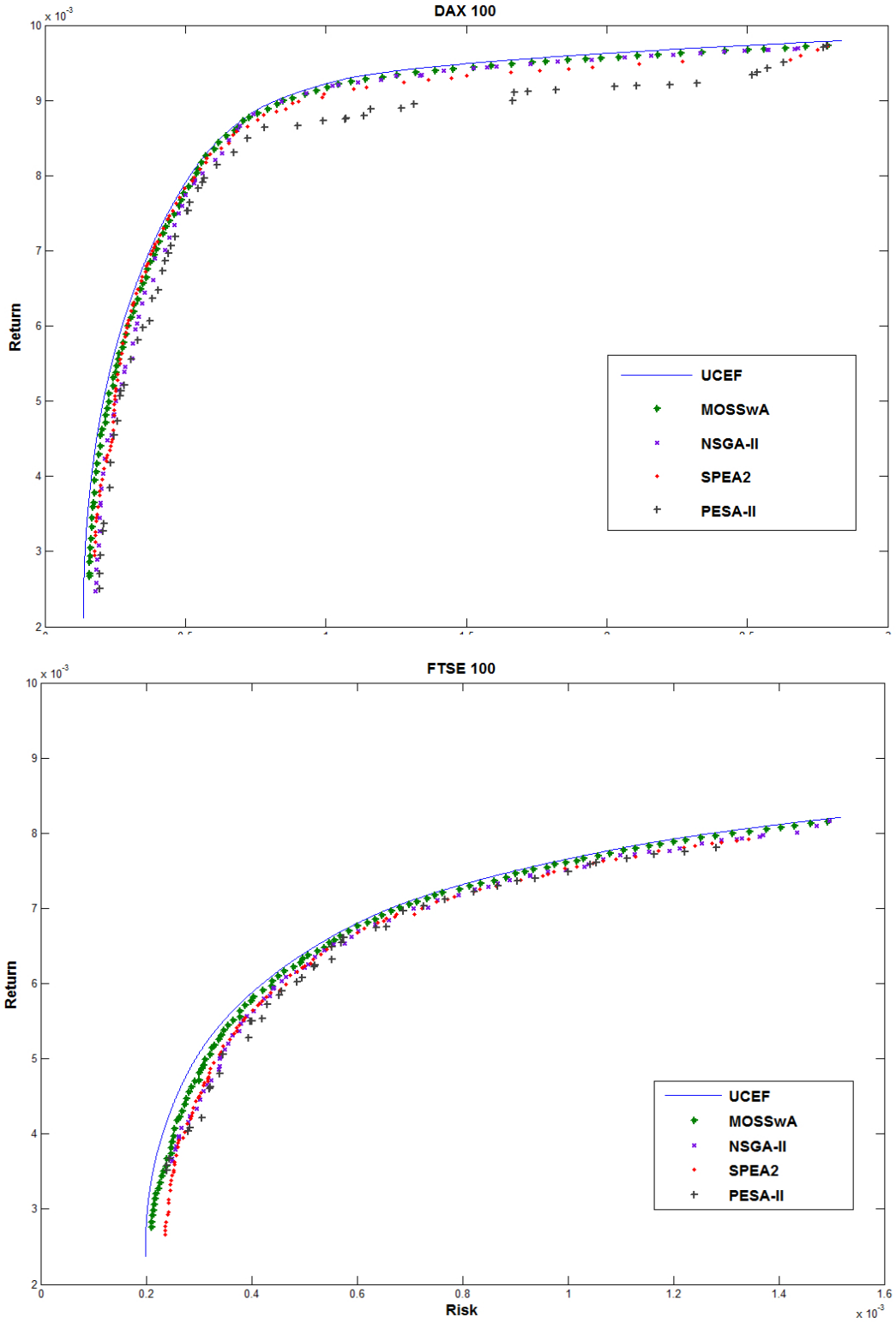


Figure 5.7: Comparison of obtained Efficient Frontier of all the algorithms for constrained portfolio optimization problem.

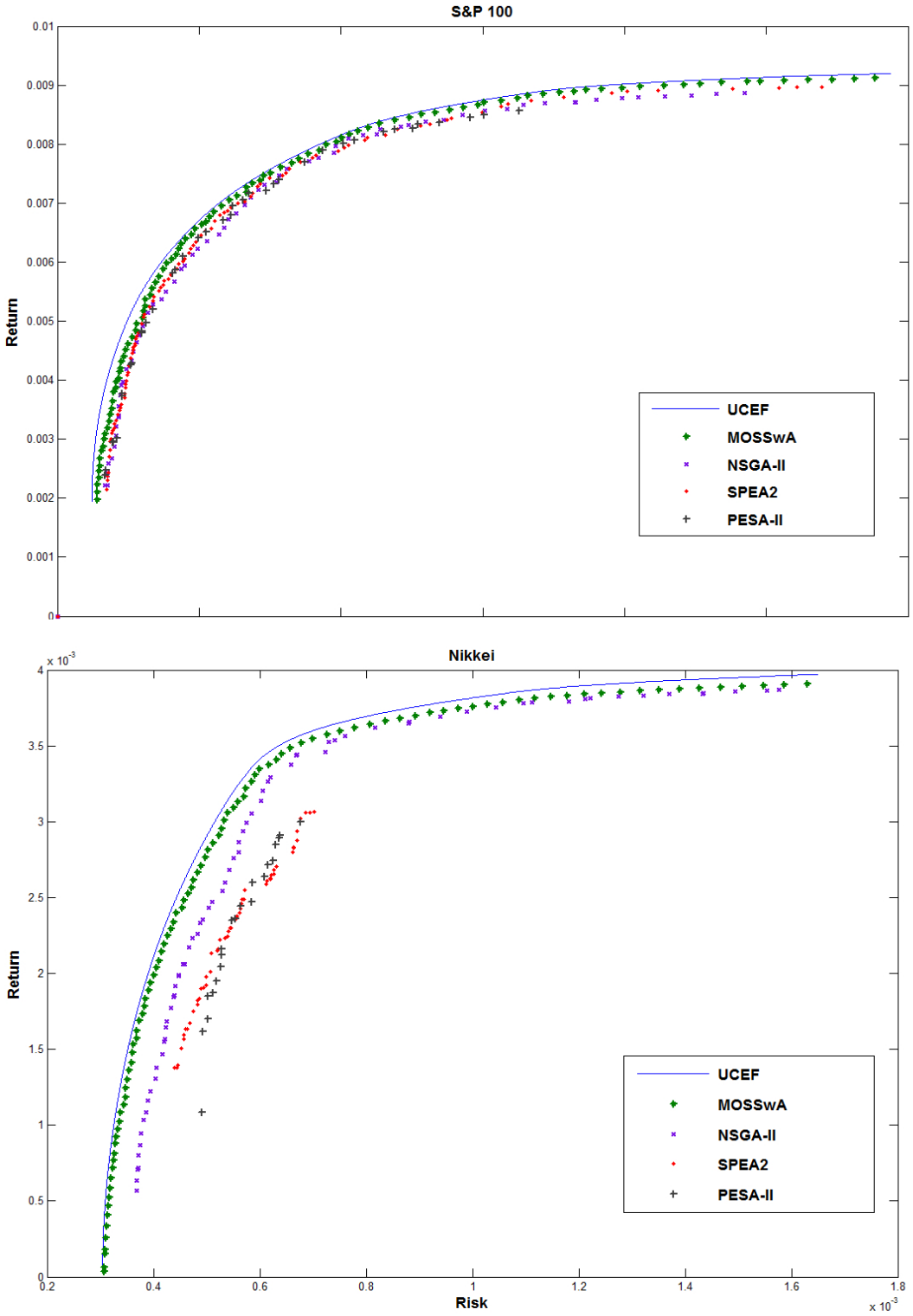


Figure 5.7: Comparison of obtained Efficient Frontier of all the algorithms for constrained portfolio optimization problem.

As stated in Section 3.4.2, IGD can provide an assessment of the overall performance of an algorithm, measuring its convergence and diversity simultaneously. We therefore compare the IGD values of the four algorithms by using Student's t-test Walpole et al. (1998). The statistical results obtained by a two-tailed t-test with 38 degrees of freedom at a 0.05 level of significance are given in Table 5.2. The result of Algorithm1 $\leftrightarrow$ Algorithm2 is shown as "$+$","$-$", or "$\sim$" when Algorithm1 is significantly better than, significantly worse than, or statistically equivalent to Algorithm2, respectively.

Algorithm1 $\leftrightarrow$ Algorithm2	Hang Seng	DAX 100	FTSE 100	S & P 100	Nikkei
MOSSwA $\leftrightarrow$ NSGA-II	$+$	$+$	$+$	$+$	$+$
MOSSwA $\leftrightarrow$ SPEA2	$+$	$+$	$+$	$+$	$+$
MOSSwA $\leftrightarrow$ PESA-II	$+$	$+$	$+$	$+$	$+$
NSGA-II $\leftrightarrow$ SPEA2	$\sim$	$\sim$	$\sim$	$\sim$	$+$
NSGA-II $\leftrightarrow$ PESA-II	$+$	$+$	$+$	$+$	$+$
SPEA2 $\leftrightarrow$ PESA-II	$+$	$+$	$+$	$+$	$+$

Table 5.2: Student t-Test Results of Different Algorithms on five problem instances from OR-Library.

Results show that MOSSwA outperforms all considered algorithms in all problem instances both in terms of solution quality and computational time. For small problem instances ($D1 - D4$), NSGA-II and SPEA2 perform similarly but SPEA2 achieves the results with higher computational cost. When the problem size becomes larger, PESA-II and SPEA2 are found to be slow in convergence and not efficient both in terms of solution quality and computation time. We could therefore conclude that the proposed MOSSwA has the best optimization performance for the portfolio optimization problem with the considered constraints.

5.6 Summary and Discussion

This chapter presents a multi-objective scatter search algorithm (MOSSwA) to solve the mean variance portfolio optimization problem with cardinality, quantity and pre-assignment constraints. The proposed MOSSwA follows the basic structure of the scatter search and defines the reference set solutions based on Pareto dominance and crowding distance measures. New subset generation and combination methods are proposed to generate efficient and diverse portfolios. Three problem-specific selection methods are introduced in order to promote the efficient convergence of the search. Hill climbing (HC) operation is adopted to search for improved portfolios. In this work, elitism is adopted by maintaining a secondary population (i.e. archive A). By exploiting the useful information in the current population and archive A, MOSSwA is able to guide the search toward the Pareto optimal set. MOSSwA significantly outperforms three existing MOEAs both in terms of distance and diversity performance measures.

Intuitively, imposing pre-assignment constraints could deteriorate the solution quality unless the pre-assigned assets belong to optimal solutions of the efficient frontier. More precisely, when a low-return security is pre-assigned the deterioration could be large whereas when a high-return asset is pre-assigned the deterioration could be less. Gaspero et al. (2011) showed that the pre-assigned assets have impact on the quality of the obtained solution. An investor is thus expected to have a good insight on setting his/her preference assets in the portfolio. In this work, the pre-assignment asset $(z_{30} = 1)$ is set by using the preliminary test results of D1 and the same pre-assignment asset is used for all five datasets (D1 $-$ D5) for consistency. We do not claim here that the pre-assignment asset used in the computational analysis belongs to optimal solutions of efficient frontier of all five datasets.

In this work we consider the cardinality constraint which limits the maximum number of assets in a portfolio. The obtained efficient frontier would be a combination of portfolios with various numbers of assets. Therefore, a portfolio manager has an extra trade-off criterion (the cardinality of the obtained solu-

tions) to take into account when making decisions to select a suitable portfolio weighting between the risk, return and the cardinality of the portfolio.

Chang et al. (2000) reasoned that the cardinality constraint with inequality case can be solved by repeatedly running the equality case incrementing the $\mathcal{K}$ values up to the maximum limit (i.e. $\mathcal{K} = 1, \ldots, \mathcal{K}_U$). The decision maker will then be better informed with several constrained efficient frontiers for different $\mathcal{K}$ values. From a computational perspective, the cardinality with equality case is far more challenging than the one with inequality case. In fact, Woodside-Oriakhi et al. (2011) investigated the cardinality constrained portfolio optimization problem with two cases using CPLEX (version 11.0) on OR-library dataset D2. The author showed that the portfolio selection problem with inequality case is computationally far easier than the equality case. Existing research also shows that majority of the studies have adopted the cardinality constraints with inequality case. Based on these observations, we are motivated to consider the strict cardinality (i.e. with equality) constraints in our next studies in the following chapters.

Chapter 6

A Learning-guided MOEA for Portfolio Optimization

> "The purpose of computation is insight, not numbers."
>
> Richard Hamming

6.1 Introduction

In this chapter, we propose a new learning-guided multi-objective evolutionary algorithm for the mean-variance portfolio optimization problem. The basic MV model is extended to consider the strict cardinality, quantity, pre-assignment and round lot constraints. We investigate the performance of the learning-guided multi-objective evolutionary algorithm with external archive (MODEwAwL) on the extended MV model with four constraints considered. Randomly generating a new candidate solution is very unlikely to achieve a good-quality practical solution for the constrained portfolio optimization problem nor to promote efficient and effective convergence. Instead, an efficient learning-guided solution generation scheme incorporating additional problem-specific heuristics is proposed to generate a good-quality solution. The proposed solution generation scheme is designed to enhance an efficient convergence of the search algorithm by concen-

trating on the promising areas of the search space.

In the development of new MOEAs, two common goals, namely fast convergence to the Pareto optimal front and good distribution of solutions along the front, are mainly considered. In order to achieve the efficient and effective convergence, a MOEA needs to be designed to exploit information accumulated about an initially unknown search space, especially in a large and complex search space. One way is to observe the interactions between the individuals and extract important features from the good solutions. A learning mechanism is introduced in an effort to identify the promising subset of assets. A subset optimization heuristic employed by Woodside-Oriakhi et al. (2011) is a key motivation for the development of a learning mechanism. This work is also motivated by the successful results of a Differential Evolution for Multi-objective Optimization (DEMO) proposed by Robič and Filipič (2005) in a wide range of applications.

6.2 Problem Model

In this work, four real-world constraints, cardinality, quantity, pre-assignment and round lot, are considered (see Section 2.4). Mathematically, the problem with considered constraints can be formulated as follows:

$$min \qquad f_1 = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} w_i w_j \sigma_{ij} \tag{6.1}$$

$$max \qquad f_2 = \sum_{i=1}^{\mathcal{N}} w_i \mu_i \tag{6.2}$$

$$subject\ to \quad \sum_{i=1}^{\mathcal{N}} w_i \leq 1 \tag{6.3}$$

$$\sum_{i=1}^{\mathcal{N}} s_i = \mathcal{K}, \tag{6.4}$$

$$w_i = y_i.\vartheta_i, \quad i = 1, ..., \mathcal{N}, \quad y_i \in \mathbb{Z}_+ \tag{6.5}$$

$$\epsilon_i s_i \leq w_i \leq \delta_i s_i, \quad i = 1, ..., \mathcal{N}, \tag{6.6}$$

$$s_i \geq z_i, \quad i = 1, ..., \mathcal{N} \tag{6.7}$$

$$s_i, z_i \in \{0, 1\}, \quad i = 1, ..., \mathcal{N} \tag{6.8}$$

where f_1 and f_2 are risk and return objectives respectively (see Section 2.2.2), $\mathcal{K}$ is the number of assets in a portfolio, binary variable s_i denotes whether asset i is invested or not. If s_i equals one, asset i is chosen to be invested and the proportion of capital w_i lies in $[\epsilon_i, \delta_i]$, where $0 \leq \epsilon_i \leq \delta_i \leq 1$. Otherwise, asset i is not invested and w_i equals zero. Eq. (6.7) defines the pre-assignment constraint to fulfil the investors' subjective requirements where the binary vector z_i denotes if asset i is in the pre-assigned set that has to be included in the portfolio or not. Eq. (6.5) defines the round lot constraint where y_i is a positive integer variable and ϑ_i is the minimum lot that can be purchased for each asset (see Section 2.4.3). As denoted in Section 2.4.3, the inclusion of the round lot constraint may make it impossible to exactly satisfy the budget constraint (see Eq. (6.3)) as the total capital might not be the exact multiples of the required trading lot for various assets.

6.3 Related Work

In this section, we review the studies which considered round lot constraints. For a comprehensive overview of the portfolio problems with different combination of constraints, we direct the interested reader to (Di Tollo and Roli, 2008; Metaxiotis and Liagkouras, 2012; Ponsich et al., 2013; Tapia and Coello, 2007).

Arriaga and Valenzuela-Rendón (2012) presented a Steepest Ascent Hill Climbing algorithm (SAHC) for portfolio selection problem with cardinality, quantity and round lots constraints. The performance of SAHC was compared with GA

and QP for three instances, NASDAQ-100, FTSE-100 and DAX-30 indices. The results showed that SAHC is competitive to GA.

Mansini and Speranza (1999) proposed three heuristic algorithms for portfolio selection problem with minimum transaction lots and tested the performance of them using real market data from the Milan Stock Exchange. The experiments showed that the proposed heuristics achieved very good solutions in a reasonable computational time. Their work also showed that the portfolio selection problem with transaction lots is an NP-complete problem (Garey and Johnson, 1990).

Soleimani et al. (2009) presented a GA with complete deterministic selection technique (i.e. the fittest half of the population survives). They considered three constraints: cardinality, minimum transaction lots and market (sector) capitalization[5]. Their work presented computational results for a number of test problems involving up to 2000 assets.

Streichert et al. (2004a,b,c) investigated the impact of binary encoding (natural binary and gray codes), a real-value encoding and hybrid encoding (i.e. binary + real). Their work considered cardinality, buy-in thresholds (i.e. floor) and transaction lots constraints. They showed that a hybrid encoding outperformed the other representations when no Lamarckism (heritability of acquired characteristics) is adopted and cardinality constraints are considered.

Lin and Liu (2008) studied the extensions of the MV model with round lots constraints and devised a GA to solve the models. In their GA, the offspring does not directly replace the parent. Instead, it replaces a randomly selected chromosome except the best one, only when it is better than the worst chromosome of the population. Computational results using 3-year Taiwanese mutual fund data showed the efficiency of GA in terms of computational time and solution quality.

[5] The sector capitalization constraints impose that some assets belong to sectors (sets of assets) and the capital invested in sector 1 is greater than the one invested in sector 2 and so on.

6.4 Learning-guided MOEA (MODEwAwL)

The multi-objective portfolio optimization problem becomes too complex to solve by numerical methods when the practical constraints reflecting investors' preferences and/or institutional trading rules are considered. Over the last few years, MOEAs have received a significant amount of attention and demonstrated their effectiveness and efficiency in solving the portfolio optimization problems with real-world constraints.

DEMO (Robič and Filipič, 2005) is one of the recent algorithms which combines the advantages of DE (Storn and Price, 1997) with the mechanisms of Pareto-based sorting and crowding distance sorting (Deb et al., 2002). It had been successfully tested on the carefully designed test functions (ZDT) introduced in (Zitzler et al., 2000). The procedure of the DEMO is described in Algorithm 6.1.

Algorithm 6.1: Procedure of DEMO (Robič and Filipič, 2005).

Input: P_{size}: the number of individuals in population P ;
Output: P;

```
P := ∅;
evaluate the initial population P of random individuals;
while stopping criterion not met do
    for each individual p_i ∈ P do
        create a candidate p' from parent p_i;
        evaluate p';
        if p' dominates p_i then
            p' replaces p_i;
        else if p_i dominates p' then
            discard p';
        else
            add p' to P;
    if | P |≥ P_size then
        truncate P;
    randomly enumerate the individuals in P;
```

DEMO maintains a population of individuals, where each represents a potential solution to the optimization problem. During the evolution, it allows its population capacity expand in order to add newly found non-dominated solutions (see Algorithm 6.1, line 3-13). Hence, it enables the newly found non-dominated solutions to immediately take part in the generation of the subsequent candidate solutions. This feature of DEMO promotes fast convergence towards the true Pareto front. In each generation, if the population exceeds the size limit, it is sorted based on the non-domination and crowding distance metrics (Deb et al., 2002) in order to identify those individuals to be truncated. It thus aims to maintain a good distribution of non-dominated solutions.

In this work, we propose a learning-guided multi-objective evolutionary algorithm (MODEwAwL) for the constrained portfolio optimization. This work has been published in Applied Soft Computing (Lwin et al., 2014). The proposed algorithm adopts a new approach to extend the generic DEMO scheme to solve the constrained portfolio optimization problem. The main differences of our approach with respect to the DEMO scheme in the literature can be outlined as follows:

1. A secondary population (i.e. an external archive) is introduced to store the well spread non-dominated solutions found throughout the evolution (see Section 6.4.8).

2. A learning mechanism is proposed to extract the important features from the efficient solutions found throughout the evolution (see Section 6.4.3).

3. An efficient solution generation scheme utilizing the learning mechanism, problem specific heuristics and effective direction-based search methods is proposed to guide the search towards the promising region of the search space (see Section 6.4.4).

The proposed MODEwAwL uses archive A to extract the important features of non-dominated solutions. Incorporating learning mechanism and prior problem-specific knowledge exploitation in the evolution process allows MODEwAwL to generate promising offspring solutions. The proposed MODEwAwL thus aims to

promote convergence by concentrating on the promising regions of the search space. The pseudocode of the proposed algorithm is described in Algorithm 6.2.

Algorithm 6.2: Procedure of MODEwAwL.

Input: $\mathcal{NP}$: the number of individuals in population P,
$\mathcal{A}_{size}$: the capacity of the archive A,
$\mathcal{CR}$: the crossover probability,
$\mathcal{F}$: a scaling factor;
Output: A;
`// Initialization (see Section 6.4.2)`
$P := \emptyset$; $A := \emptyset$;
$P \leftarrow$ randomly create an initial population;
while *stopping criteria not met* **do**
  `// Update the archive` A `(see Section 6.4.8)`
  $A \leftarrow$ *update* the archive A with non-dominated solutions from P;
  if $\mid A \mid \geq \mathcal{A}_{size}$ **then**
    $A \leftarrow$ maintain A with $\mathcal{A}_{size}$ least crowded non-dominated solutions;
  `// Learning mechanism (see Section 6.4.3)`
  learn from the archive A to identify the promising asset(s);
  for ***each*** *individual* $p_i (i = 1, \ldots, \mathcal{NP})$ *in* P **do**
    `// Candidate generation (see Section 6.4.4)`
    $p' \leftarrow$ *create* new candidate p' from P and learning mechanism;
    if *candidate* p' *does not satisfy constraints* **then**
      repair p' (see Section 6.4.5);
    evaluate p' by f_1 and f_2 (see Eqs. 6.1 - 6.2);
    if p' *dominates* p_i **then**
      p' replaces p_i;
    else if p_i *dominates* p' **then**
      discard p';
    else
      add p' to the current population P;
  if $\mid P \mid \geq \mathcal{NP}$ **then**
    `// Truncate` P `(see Section 6.4.7)`
    $P \leftarrow$ maintain P with best $\mathcal{NP}$ solutions, ranked by non-domination and crowding distance metrics;
  randomly enumerate the individuals in P;

6.4.1 Solution representation and encoding

In our solution representation, we adopt the same hybrid encoding used in Section 4.4.1. Some existing research studies (Anagnostopoulos and Mamanis, 2011b; Mishra et al., 2014; Skolpadungket et al., 2007; Streichert et al., 2004a,b,c) adopt similar encoding to define a portfolio. When the cardinality and pre-assignment constraints are considered, the introduction of binary variables s_i in the multi-objective portfolio model enhances the evaluation of the algorithm.

6.4.2 Initial population generation

To generate an initial population, $\mathcal{K}$ different assets (including all assets in the pre-assignment subset) are randomly selected and proportions are assigned to those selected assets randomly. If the generated portfolio violates the budget and/or quantity constraints, such solution is corrected by the constraint handling techniques provided in Section 6.4.5. Hence, all generated solutions in the trial population are feasible.

6.4.3 Learning mechanism

At each generation, the distribution of assets from non-dominated solutions in the external archive is observed to identify the promising assets. The concentration score of each asset c_i is calculated by counting its occurrences in the archive divided by the archive size.

$$c_i = \frac{\sum_{j=1}^{|A|} s_{i,j}}{\mid A \mid}.$$

The new solutions to be generated are encouraged to be composed of those assets by exploiting the knowledge obtained throughout the evolution to direct the search towards the promising search space (see Section 6.4.4). The proposed learning mechanism is computationally cheap as it only uses a single update at

each generation. A similar form of scoring function has been used as one of the components in the trade-off studies by Smith et al. (2007).

6.4.4 Candidate generation

One of the factors to consider in designing the portfolio model in the proposed MODEwAwL is to find an effective way to generate offsprings. The no free lunch theorem by Wolpert and Macready (1997) states that "*the average performance of any pair of algorithms across all possible problems is identical*." Ho and Pepyne (2002) restates that "*a general-purpose universal optimization strategy is theoretically impossible, and the only way one strategy can outperform another is if it is specialized to the specific problem under consideration*."

The insights from the no free lunch theorem for optimization (Wolpert and Macready, 1997) have highlighted the need to embed domain knowledge into an EA to achieve good performance (Bonissone et al., 2006). The introduction of problem-specific knowledge in the design of the algorithm is crucial in order to perform better than random search. We aim to find effective and efficient scheme with a good balance between the exploitation and exploration. The new solution is generated by two phases: the selection of assets from a universe of $\mathcal{N}$ available assets and the allocation of capital to those selected assets. The idea presented here is to use DE for exploring the real decision variables and exploit learning mechanism and problem specific heuristics described below to select the promising assets in the new solution.

The information about the concentration of the assets in the non-dominated portfolios in the archive is exploited in selecting the promising assets for the new candidate portfolio. Hence the assets are ranked according to their concentration scores in the archive non-dominated solutions (see Section 6.4.3). The assets which score greater than zero are considered to be promising ones. The higher the score of the asset, the higher its chances to be included in the new candidate portfolio.

In the standard portfolio theory, when the prices of two assets move towards the same direction, they are said to be positively correlated. If they move in opposite directions, they are said to be negatively correlated. If the assets are independent, the covariance value will be zero or near zero. It should be observed that positively correlated assets tend to increase the resulting variance and hence the risk. Negatively correlated assets, on the other hand, tend to reduce the overall variance and thus the risk. Therefore, we can promote the chances of reducing the overall portfolio's risk if we can identify the negatively correlated assets to some extent. Table 6.1 summarizes how correlation effects co-movement of assets and risk (Israelsen, 2007; McDonnell, 2008).

Correlation	Co-movement	Effect on Risk
Positive	Together	Minimal decrease
Negative	Opposite	Large decrease
Zero	Independent	Moderate decrease

Table 6.1: How correlation effects co-movement of assets and risk.

In order to generate a new candidate solution, the $\mid Z \mid$ assets are firstly selected if the pre-assignment constraint is considered. By taking into account of the above stated intuitive learning, in this work, the proposed MODEwAwL then alternatively uses the following selection schemes to fill the remaining assets:

S1: $\mathcal{K} - \mid Z \mid$ assets are selected by roulette wheel selection based on the concentration score c_i.

S2: $\mathcal{K} - \mid Z \mid$ assets which have the highest concentration score c_i are selected.

S3: $\mathcal{K} - \mid Z \mid$ assets which have the highest expected return values are selected.

S4: A random n number of assets (where $n = randint[\ 0, \mathcal{K}- \mid Z \mid]$) which have the highest concentration score c_i are selected. The remaining ($\mathcal{K} - n- \mid Z \mid$) assets are filled by randomly selecting one of the following methods:

- Select those assets which have the lowest risk values.
- Select those assets which have the highest return values (i.e. **S3**).
- Select those assets which have the least correlation from those n assets already chosen.

By adopting the above stated selection scheme, the new candidate solution satisfies the pre-assignment and cardinality constraints. The proportions of those selected assets for the new candidate solution are assigned by using a direction-based offspring generation scheme where $p1$, $p2$ and $p3$ (with $w1_i$, $w2_i$ and $w3_i$ allocations respectively) are randomly selected portfolios from the current population P as follows:

W1: $w'_i := w3_i + rand[0,1] \times (w1_i - w2_i)$

W2: $w'_i := w3_i + \mathcal{F} \times (w1_i - w2_i)$

W3: rank $p1$, $p2$ and $p3$ by dominance and crowding distance measure (i.e., $p1$ is the best portfolio and $p3$ is the worst portfolio among three portfolios) and generate weight allocations of candidate portfolio by directing away from $p3$ and towards the middle between $p1$ and $p2$ as follows:

$$w'_i := (w1_i + w2_i)/2$$

The detailed procedure of the candidate generation is provided in Algorithm 6.3. The proposed candidate generation mechanism intends to guide the search toward promising direction by learning from the reference assets from the archive and reference proportions from the current population. In this way, the proposed algorithm converges efficiently. The new candidate portfolio is repaired if

the quantity and round lot constraints are violated (see the repair mechanism in Section 6.4.5).

Algorithm 6.3: Candidate Generation of MODEwAwL.

Input: concentration scores of assets $c_i, i = 1, \ldots, \mathcal{N}$ and a parent $\bar{p} \in P$;
Output: candidate solution p';

```
randomly select 𝒦 assets by S1, S2, S3 or S4;
randomly select an index i from those selected 𝒦 and assign i to j and χ;
randomly select three different portfolios: p1, p2, p3 ∈ {P \ p̄};
for each selected asset do
    if rand(0,1) < 𝒞ℛ or j == χ then
        allocate weight w' by W2, W1 or W3;
    else
        assign weight w' with w̄_i of parent portfolio p̄;
    if w' < ε_i then
        w' ← rand(ε_i, δ_i);
    randomly select an index i from the 𝒦 selected and assign i to j;
```

6.4.5 Constraint handling

When using an evolutionary algorithm to solve constrained optimization problems, there are various methods proposed in the literature for handling constraints in evolutionary optimization, such as penalty function method, special representations and operators, repair methods and multi-objective methods (Coello, 2002). Among those methods, repair method is one of the effective approaches to locate feasible solutions (Mezura-Montes, 2009). During the population sampling, each constructed individual portfolio is repaired if it does not satisfy all considered constraints. As described in Section 6.4.4, the new solution generated by our proposed MODEwAwL already satisfies the cardinality and pre-assignment constraints.

Hence, the following repair mechanism is applied (Skolpadungket et al., 2007; Streichert et al., 2004c) :

1. All weights of selected assets in the candidate solution are adjusted by setting $w'_i = \epsilon_i + \frac{w'_i - \epsilon_i}{\sum(w'_i - \epsilon_i)}$.
2. The weights are then adjusted to the nearest round lot level by setting $w'_i = w'_i - (w'_i \ \ mod \ \ \vartheta_i)$. The remaining amount of capital is redistributed in such a way that the largest amount of $(w'_i \ \ mod \ \ \vartheta_i)$ is added in lot of ϑ_i until all the capital is spent.

6.4.6 Selection scheme

The proposed MODEwAwL applies the elitist selection scheme based on Pareto optimality (see Algorithm 6.2). During the evolution, the population is extended by adding the newly found non-dominated solutions. Hence, at each generation, the number of portfolios in the current population will be between $\mathcal{NP}$ and $2\mathcal{NP}$.

6.4.7 Truncate population

In each generation, if the number of portfolios in the current population exceeds its limit $\mathcal{NP}$, it needs to identify those which need to be removed. The individuals in the population are sorted based on the non-dominance and crowding distance measures (Deb et al., 2002). Then the current population is truncated by keeping the best $\mathcal{NP}$ individuals for the next generation.

6.4.8 Maintaining the external archive

The main objective of the external archive A is to keep all the non-dominated solutions encountered along the search process. This approach is adopted in order to save and update all well spread non-dominated solutions generated by the algorithm during the search. In each generation, the archive A is updated with the non-dominated solutions from the trial population.

6.5 Performance Evaluation

6.5.1 Effectiveness of candidate generation and archive

In this section, our experiments focus on the impact of the learning-guided solution generation mechanism. In order to evaluate the performance of the MODEwAwL, we compare it with two variants of the algorithm: the multi-objective differential evolution (MODE) and the multi-objective differential evolution with archive (MODEwA). Figure 6.1 shows the comparisons of the three algorithms in terms of IGD, GD and Δ. The experimental results distinctly show that the proposed algorithm with the learning-guided solution generation mechanism outperforms MODE and MODEwA in most instances.

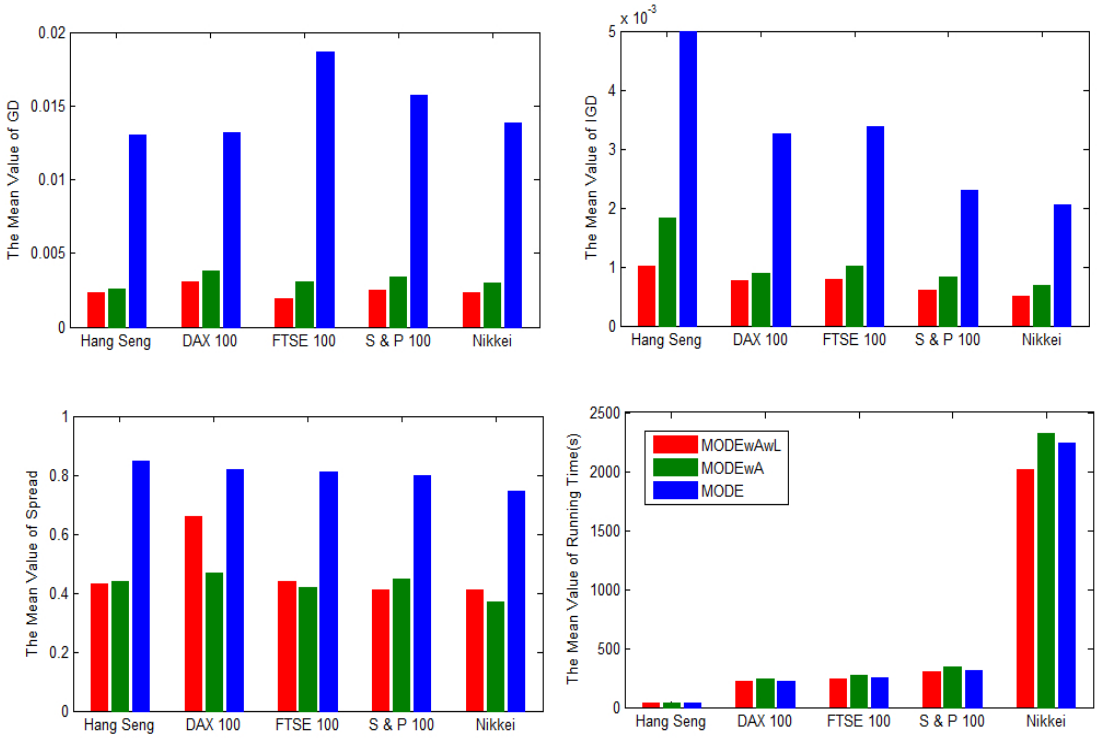


Figure 6.1: Effectiveness of the learning-guided solution generation scheme and archive.

6.5.2 Comparisons of the algorithms

In order to evaluate the overall performance of the proposed MODEwAwL, we compare it with four state-of-the-art multi-objective evolutionary algorithms in the literature.

- NSGA-II: the Non-dominated Sorting Genetic Algorithm II proposed by Deb et al. (2002). The algorithm uses binary tournament selection based on the crowding distance. It performs crossover and mutation by simulated binary crossover and polynomial mutation operators (see Section 3.3.2.1).
- SPEA2: the Strength Pareto Evolutionary Algorithm proposed by Zitzler et al. (2001). The algorithm employs fine-grained fitness assignment, density estimation techniques and archive truncation methods. Like NSGA-II, it uses binary tournament selection, simulated binary crossover and polynomial mutation evolutionary operators (see Section 3.3.2.2).
- PESA2: the Pareto Envelope-based Evolutionary Algorithm proposed by Corne et al. (2001). The algorithm uses hyper-boxes to assign fitness and employs the simulated binary crossover and polynomial mutation operations (see Section 3.3.2.3).
- PAES: the Pareto Archived Evolution Strategy proposed by Knowles and Corne (2000). The algorithm uses a simple (1+1) local search evolution strategy. It maintains an archive of non-dominated solutions and it exploits those Pareto solutions to estimate the quality of new solutions (see Section 3.3.2.4).

In order to ensure a fair comparison, we have used the same population size and archive size (if applicable) for all the algorithms tested in this work. We have chosen to run all the algorithms run for the same stopping criteria (i.e. the same number of evaluations) to generate the Pareto front. All algorithms considered in this study were coded in C# using the information from the relevant papers and run on a personal computer Intel(R) Core(TM)2 Duo CPU E8400 3.16 GHz. Each algorithm also uses the same encodings (see Section 6.4.1)

and repair mechanism (see Section 6.4.5) when a newly constructed portfolio violates the considered constraints. Before the experiments were performed, parameters were tuned for all algorithms using the smallest problem instance, i.e. Hang Seng. Table 6.2 shows the best parameter values of the algorithms.

Parameters	MODEwAwL	NSGA-II	SPEA2	PESA-II	PAES
Number of population ($\mathcal{NP}$)	100	100	100	100	100
Number of generation	1,000$\mathcal{N}$	1,000$\mathcal{N}$	1,000$\mathcal{N}$	1,000$\mathcal{N}$	1,000$\mathcal{N}$
Scaling factor ($\mathcal{F}$)	0.3	–	–	–	–
Crossover probability ($\mathcal{CR}$)	0.9	0.9	0.9	0.9	–
Crossover distribution index	–	20	20	20	–
Mutation probability	–	$1/\mathcal{N}$	$1/\mathcal{N}$	1/N	$1/\mathcal{N}$
Mutation distribution index	–	20	20	20	20
Tournament round	–	–	1	–	–
Number of bisection	–	–	–	5	5
Archive size ($\mathcal{A}_{size}$)	100	–	100	100	100

Table 6.2: Parameter setting of five algorithms.

Four performance metrics, IGD, GD, Diversity (Δ) and HV, are used to evaluate the quality of the solutions achieved by the tested algorithms (see Section 3.4). The true Pareto front for highly constrained multi-objective portfolio optimization problem considered in this work is unknown. We use the best known unconstrained efficient frontier (UCEF) provided by the OR-library (Beasley, 1990, 1999) as the true Pareto front reference set. This has been widely adopted in the literature.

In this section, a number of experiments are performed. The results of GD, IGD, Diversity (Δ) and running time of the five algorithms performed on seven

datasets (D1 − D7) from OR-library are shown in Figures [6.2 − 6.8]. These results are obtained for the constrained portfolio optimization problem with cardinality $\mathcal{K} = 10$, floor $\epsilon_i = 0.01$, ceiling $\delta_i = 1.0$, pre-assignment $z = \{30\}$ and round lot $\vartheta_i = 0.008$.

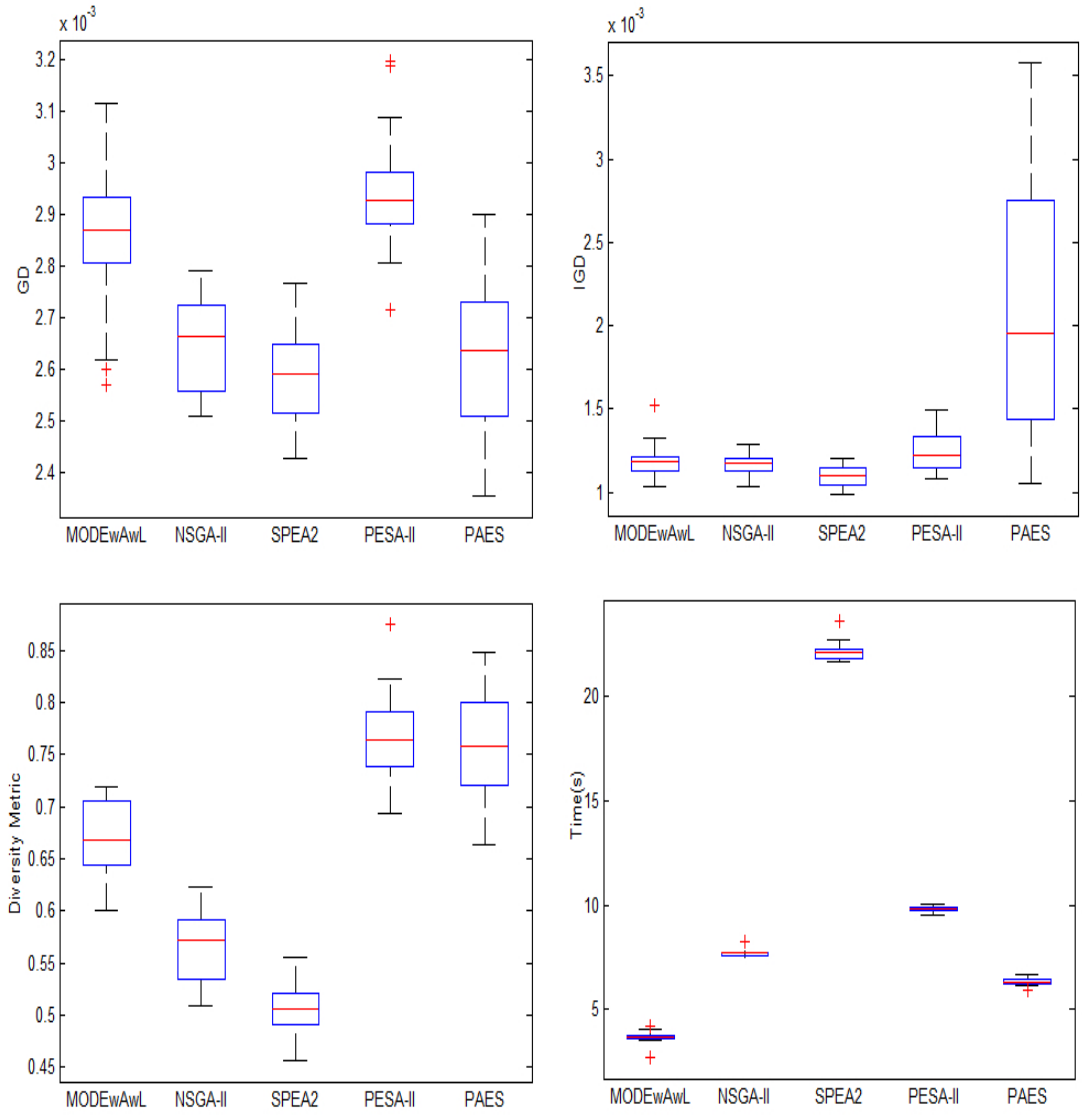


Figure 6.2: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Hang Seng dataset.

In the literature, many studies have used the cardinality value $\mathcal{K} = 10$ in their experiments. Therefore, additional experiments are also performed for different cardinality values with $\mathcal{K} = 15$ and $\mathcal{K} = 5$ but used the same parameter values for other constraints. These additional results are provided in Appendix A.

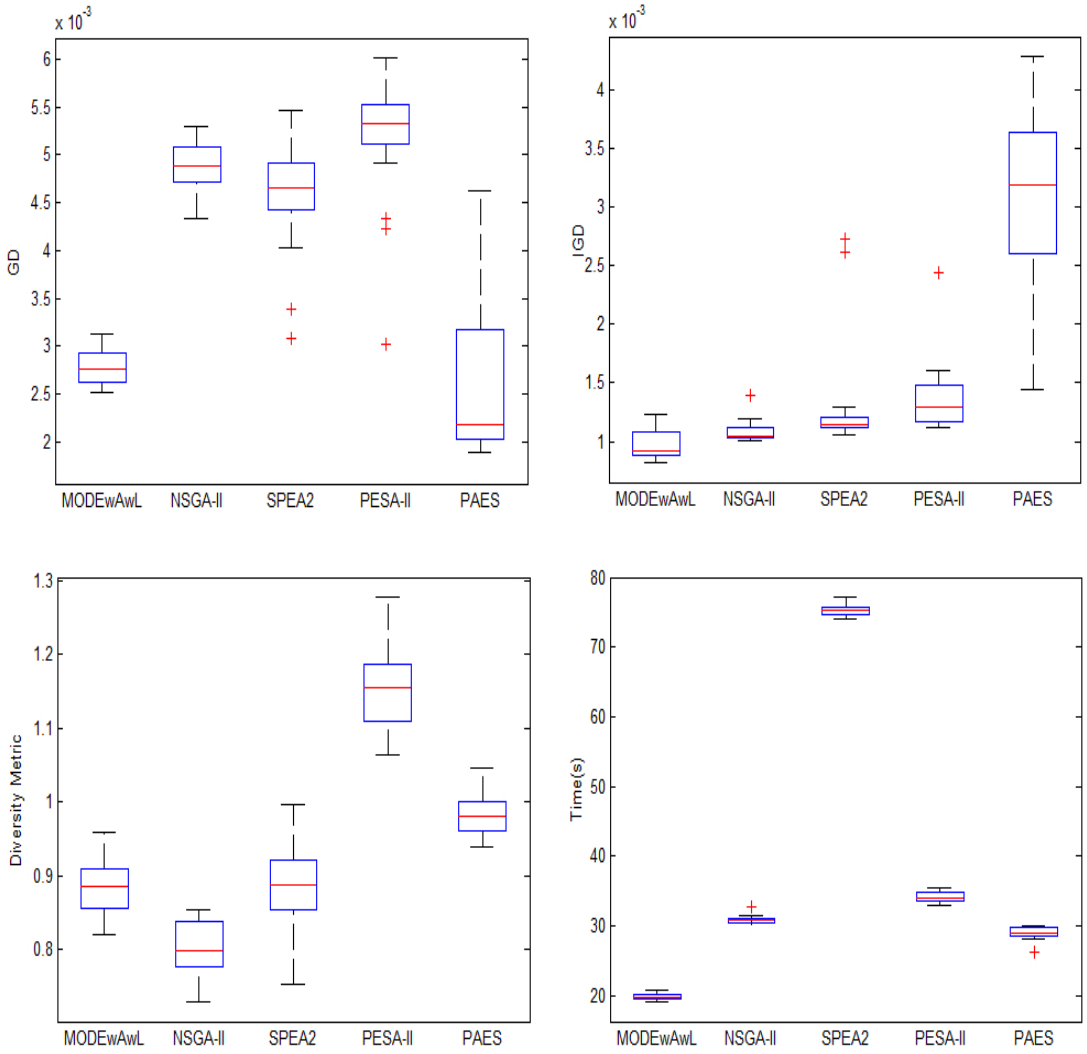


Figure 6.3: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for DAX 100 dataset.

Computational results show that for most of the problem instances (D2 – D7), the MODEwAwL obtains the smallest mean values for IGD metrics. SPEA2 and NSGA-II are comparable in most problem instances while SPEA2 achieves the results with expensive computing time. The results also confirm that PAES ranks the worst among the test algorithms for the problem considered. However, PAES is the second fastest algorithm after MODEwAwL.

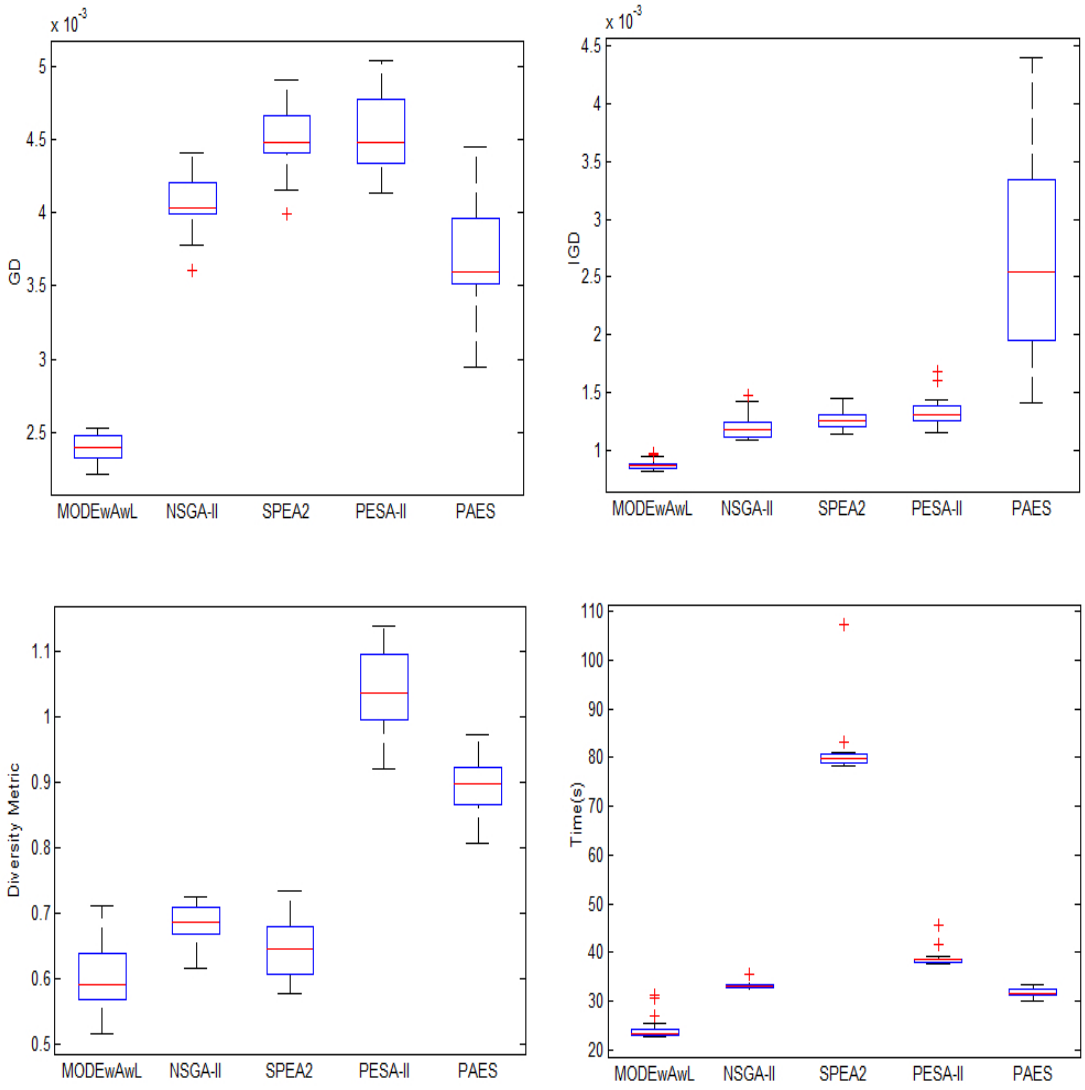


Figure 6.4: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for FTSE 100 dataset.

In terms of GD and Δ metrics, MODEwAwL performs better when the problem size becomes larger and the cardinality constraint is set 10 and 15. For computational results with $\mathcal{K} = 5$, MODEwAwL outperforms all other MOEAs in all seven problem instances.

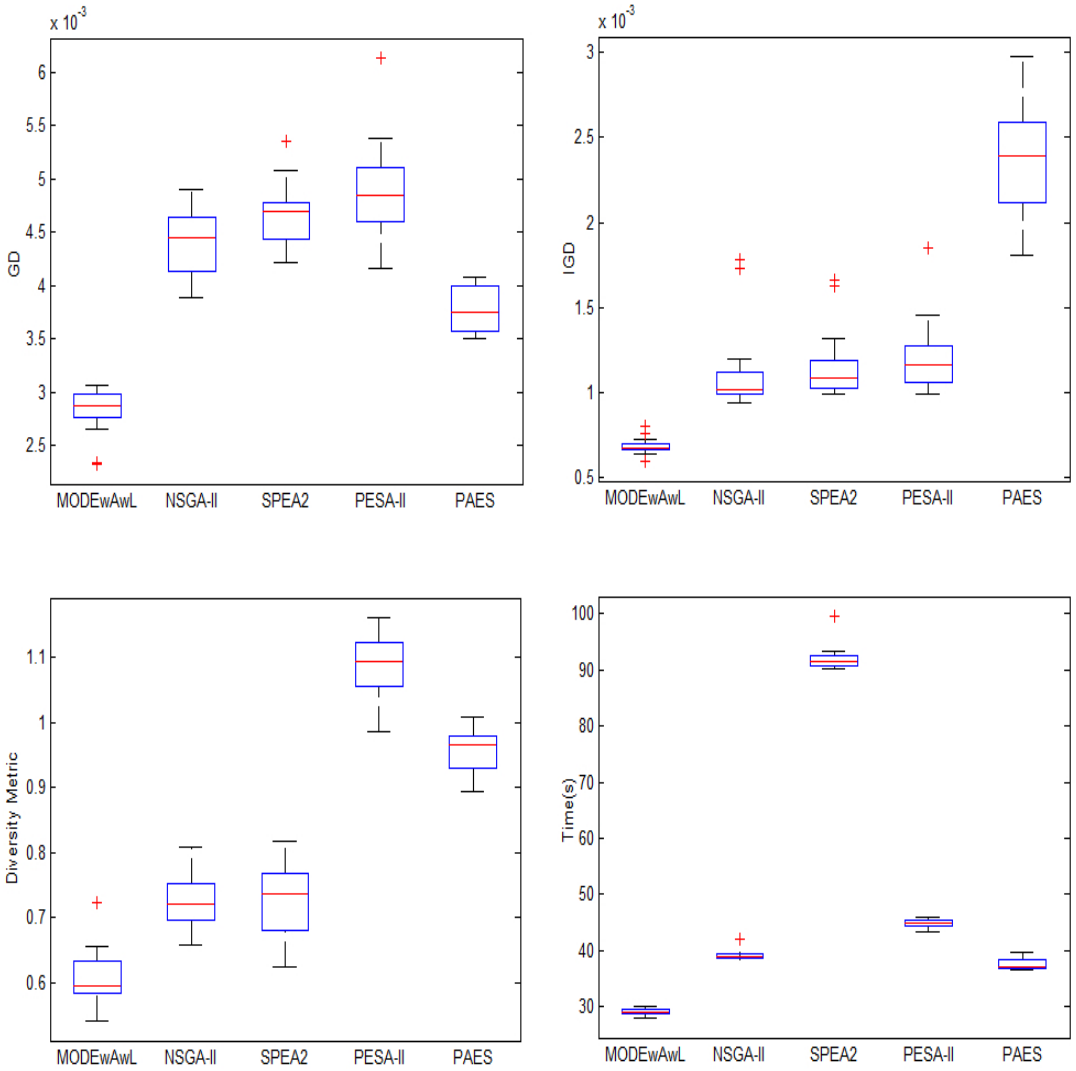


Figure 6.5: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for S & P 100 dataset.

Overall, we could conclude that MODEwAwL is significantly better than the other compared MOEAs for the majority of datasets. The experimental results provided in Appendix A have further demonstrated that the proposed algorithm is efficient for various search spaces with different values of $\mathcal{K}$. The proposed MODEwAwL is thus more robust than the compared MOEAs.

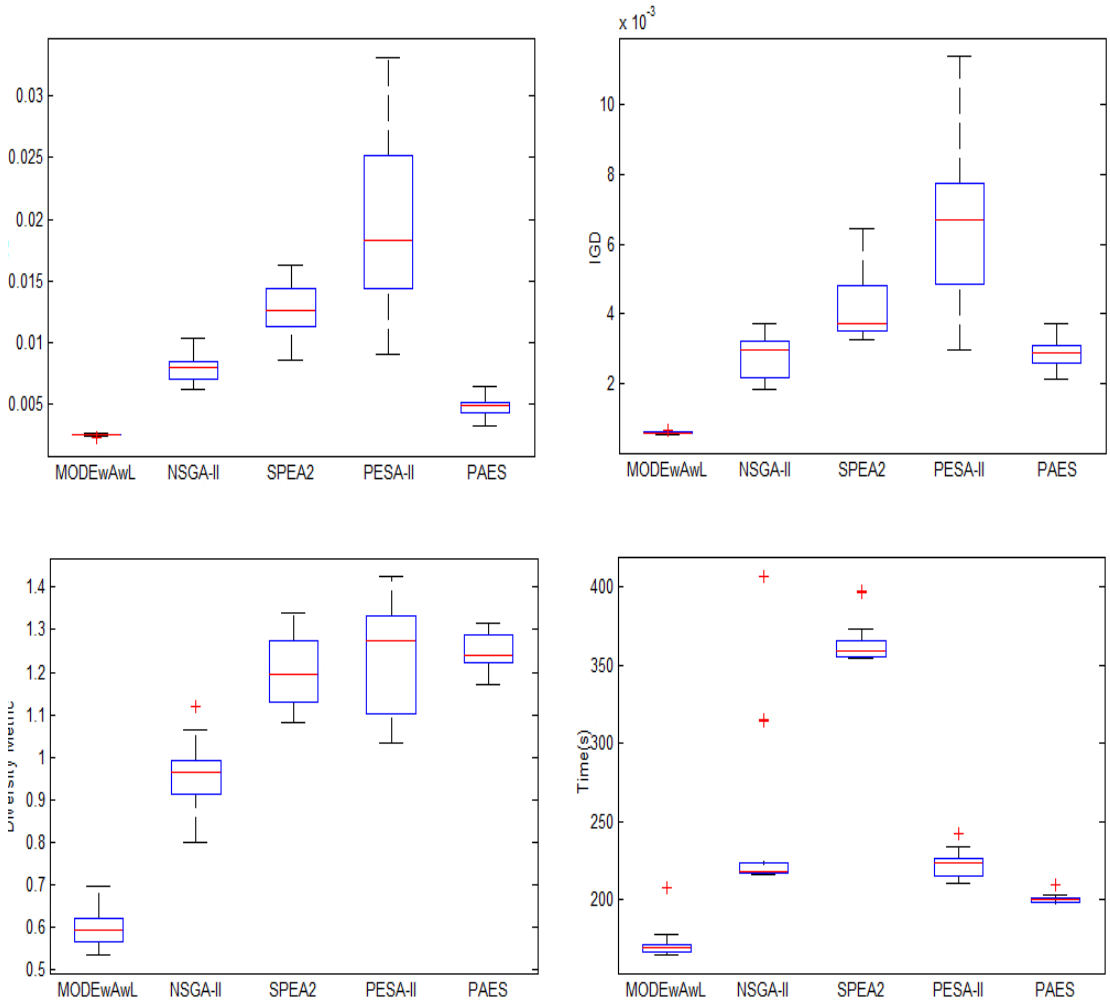


Figure 6.6: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Nikkei dataset.

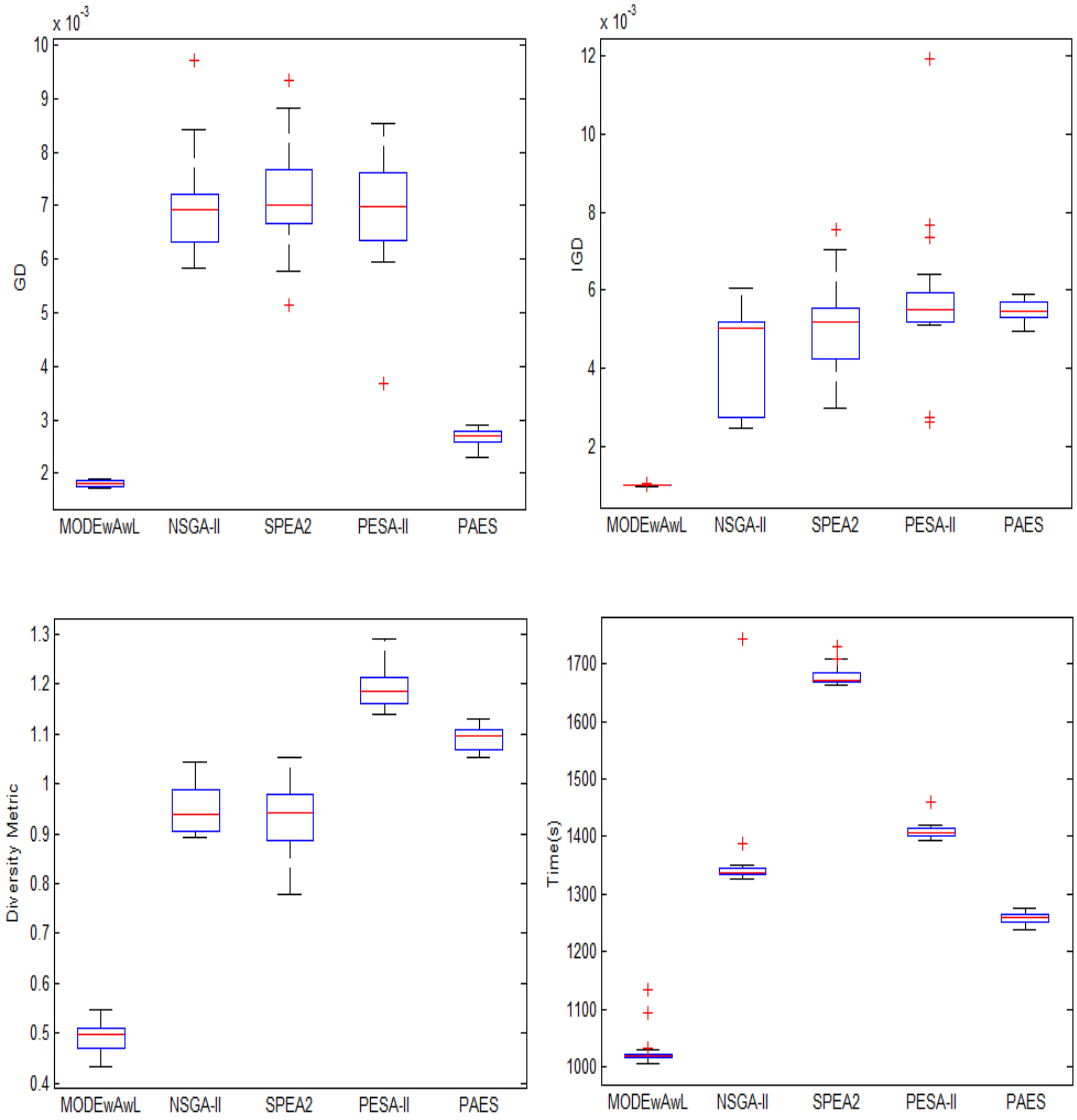


Figure 6.7: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for S & P 500 dataset.

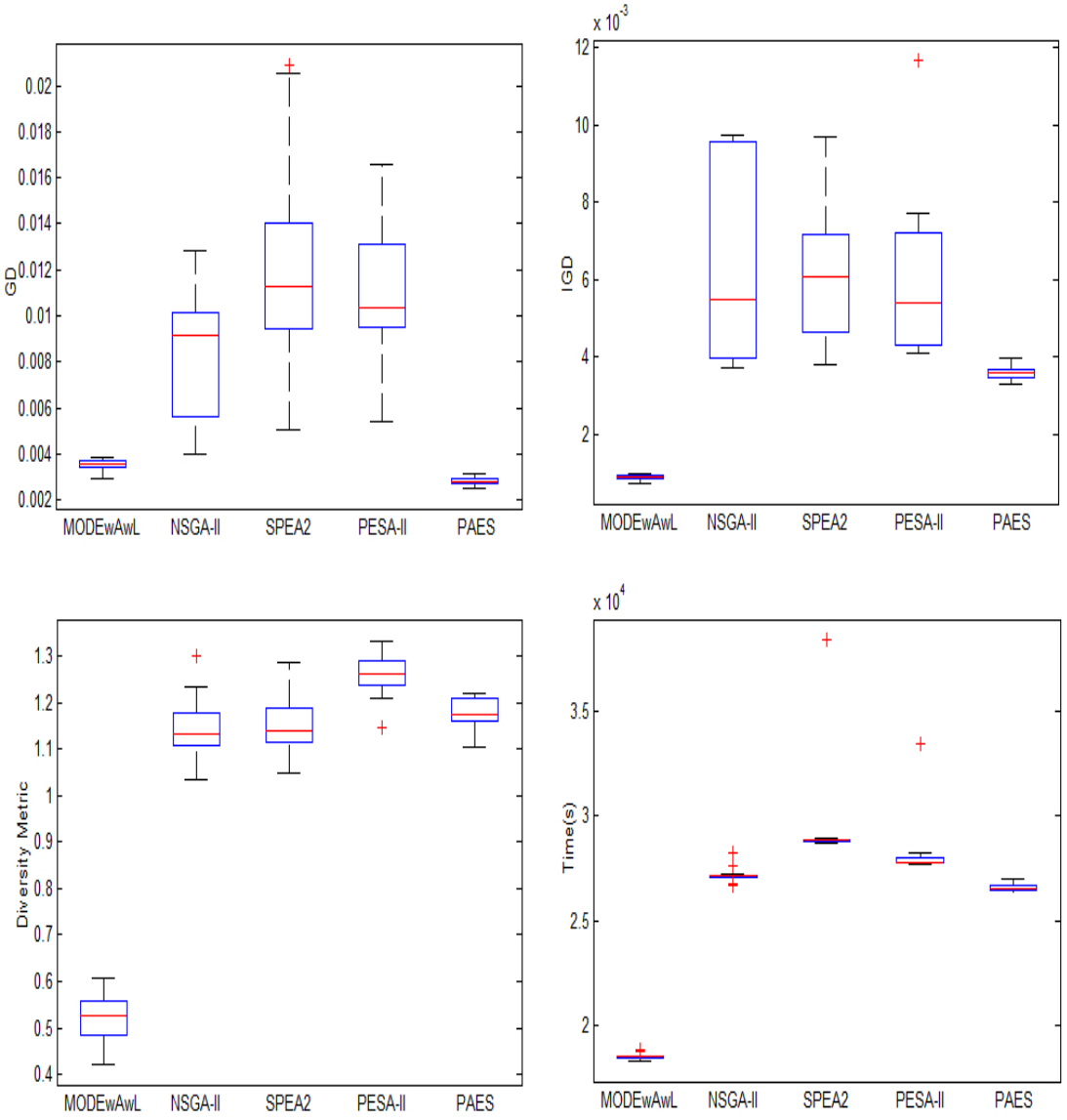


Figure 6.8: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Russell 2000 dataset.

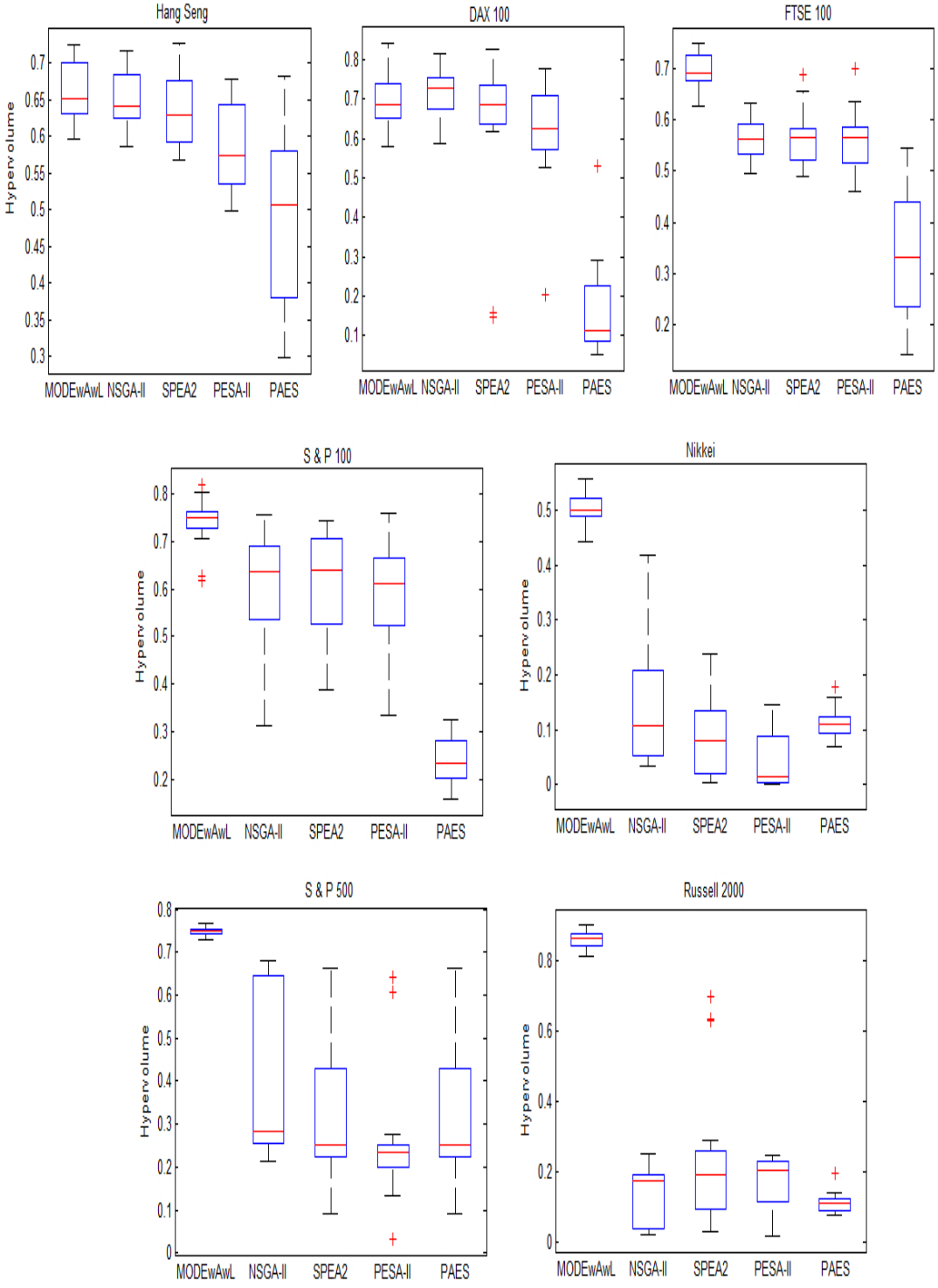


Figure 6.9: Performance comparisons of five algorithms in terms of HV metric.

Figure 6.9 shows the hypervolume (HV) calculation performed on seven datasets and for each problem instance, the results reconfirm the superiority of MODEwAwL since it outperforms in six out of seven datasets. For illustrative purpose, the obtained efficient frontiers of the algorithms for seven instances along with the true unconstrained efficient frontier (UCEF) are provided in Figure 6.10. When the problem sizes are small, the Pareto sets obtained by the considered algorithms are very competitive to each other such that it would be hard to differentiate visually. As the problem sizes increase, the proposed algorithm obtained significantly better efficient frontier than those obtained by other MOEAs considered in this work. Based on the analysis, we conclude that the proposed MODEwAwL is able to solve large-scale real-world portfolio optimization efficiently. The results also demonstrate that NSGAII and SPEA2 loose their effectiveness when the problem dimension increases.

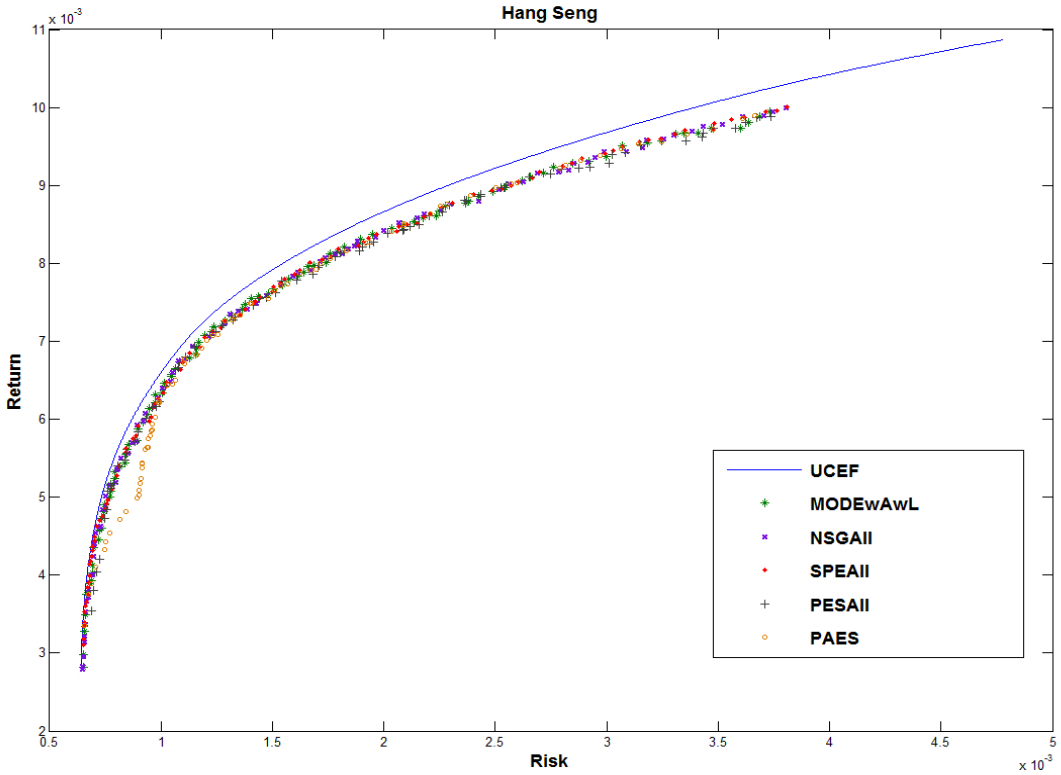


Figure 6.10: Comparison of efficient frontiers for seven datasets.

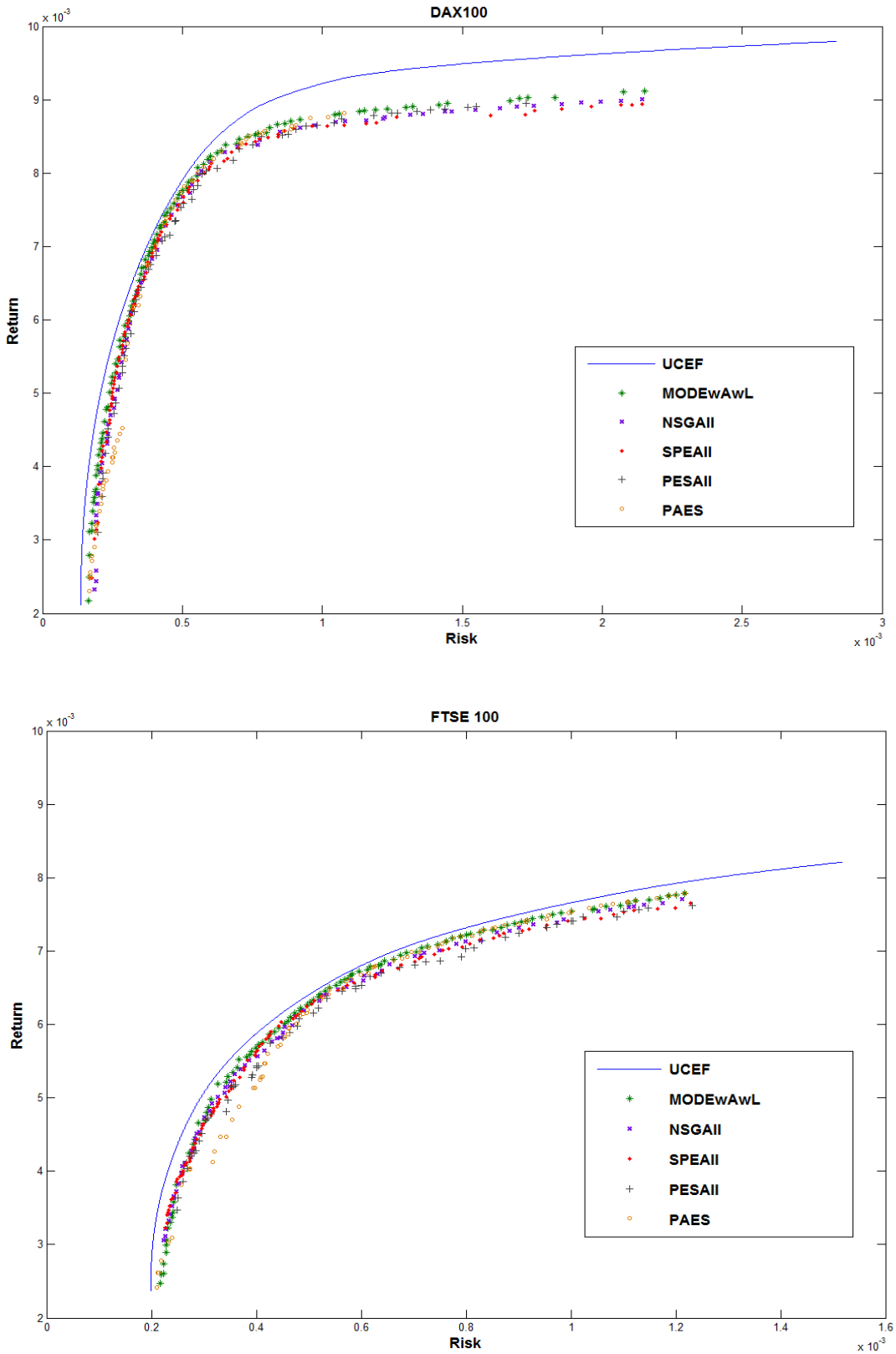


Figure 6.10: Comparison of efficient frontiers for seven datasets.

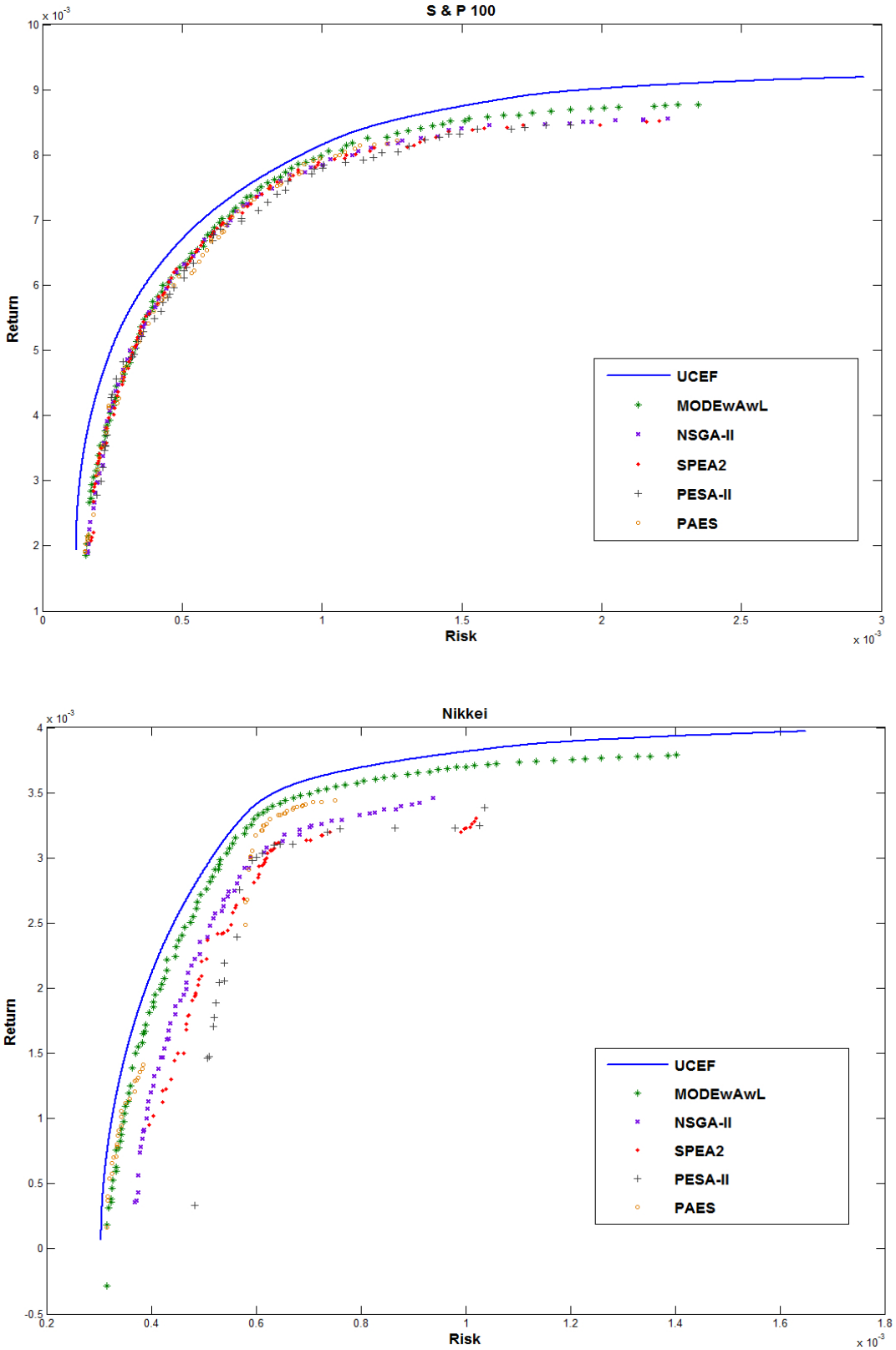


Figure 6.10: Comparison of efficient frontiers for seven datasets.

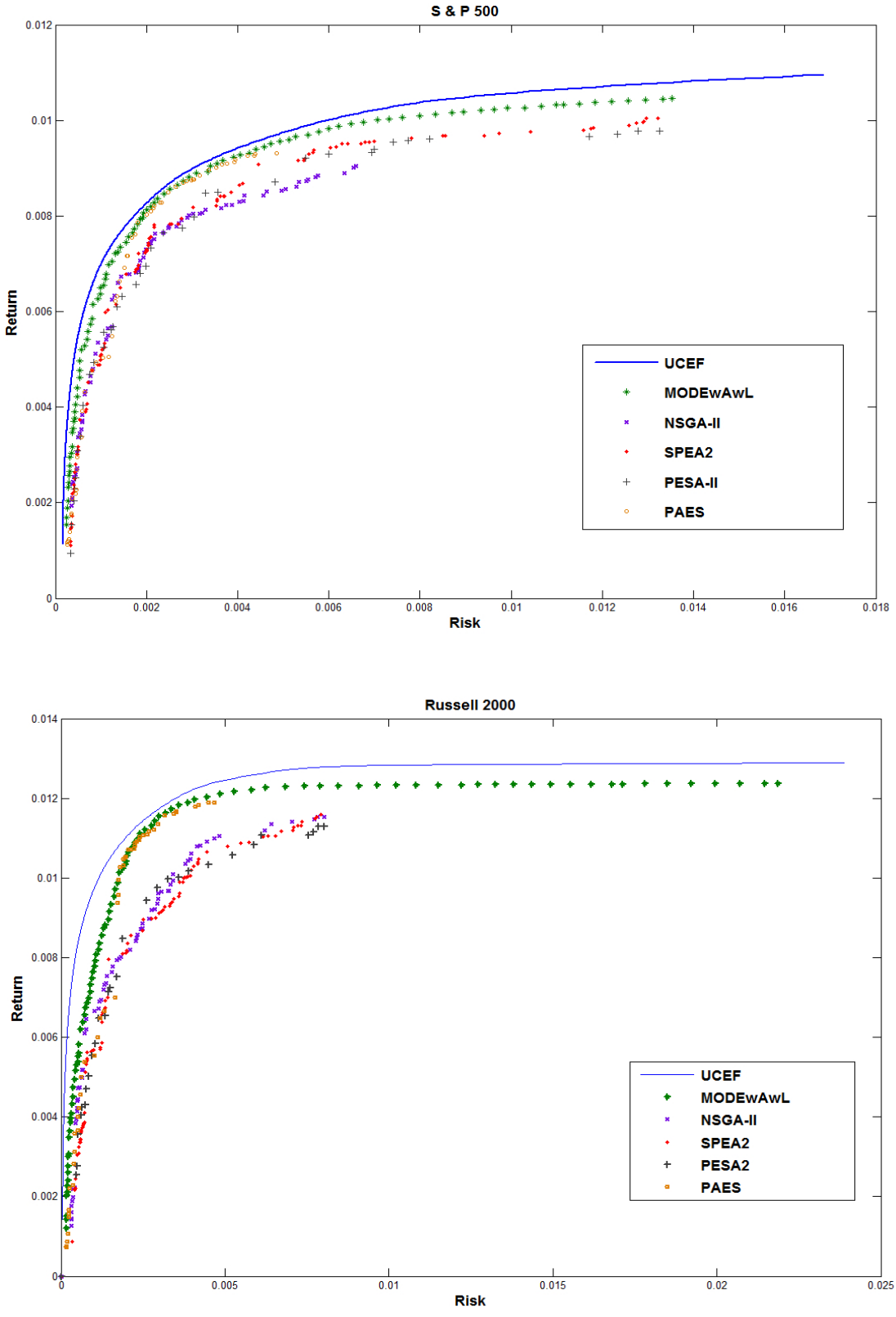


Figure 6.10: Comparison of efficient frontiers for seven datasets.

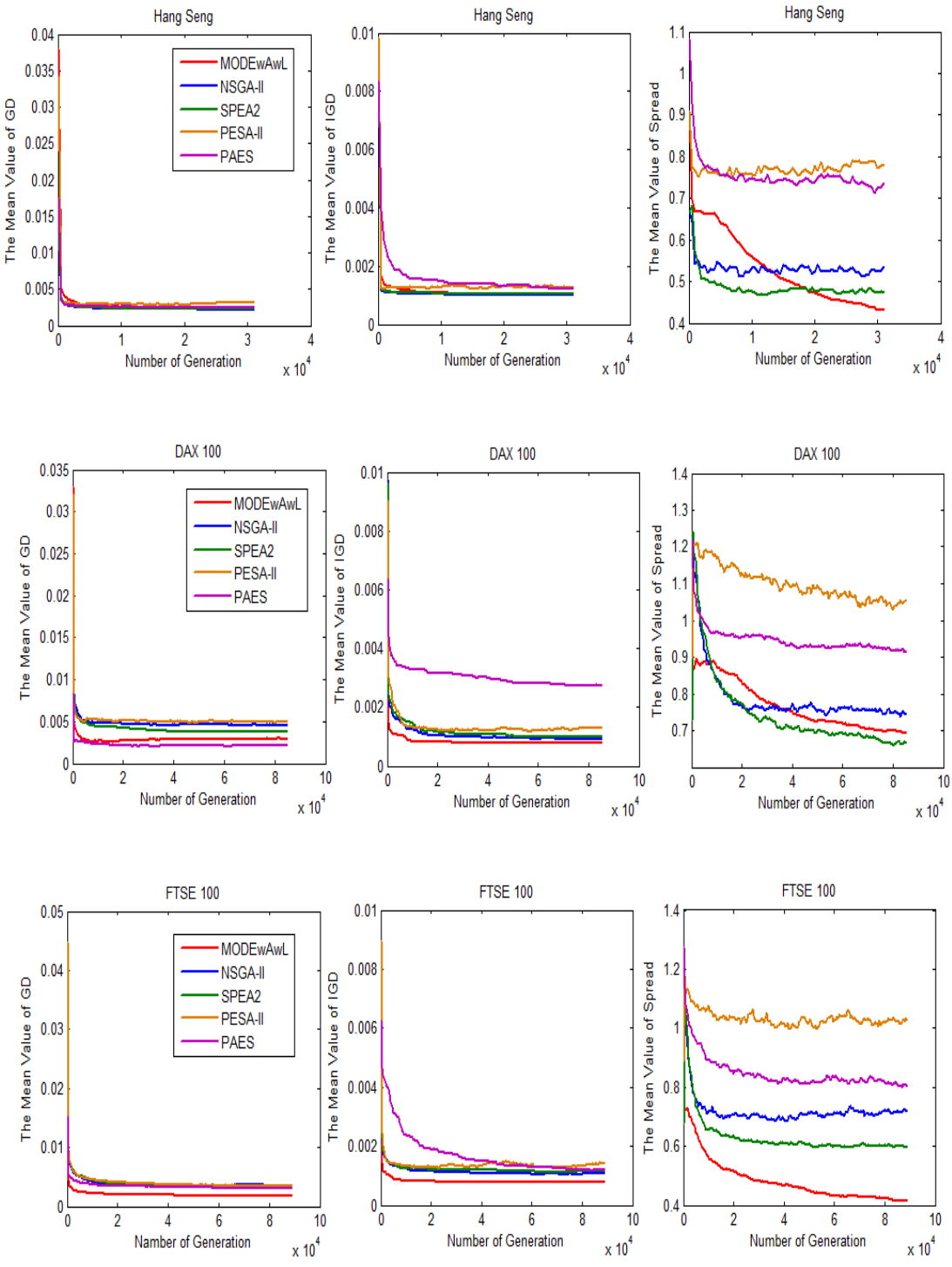


Figure 6.11: Comparisons of convergence of five algorithms.

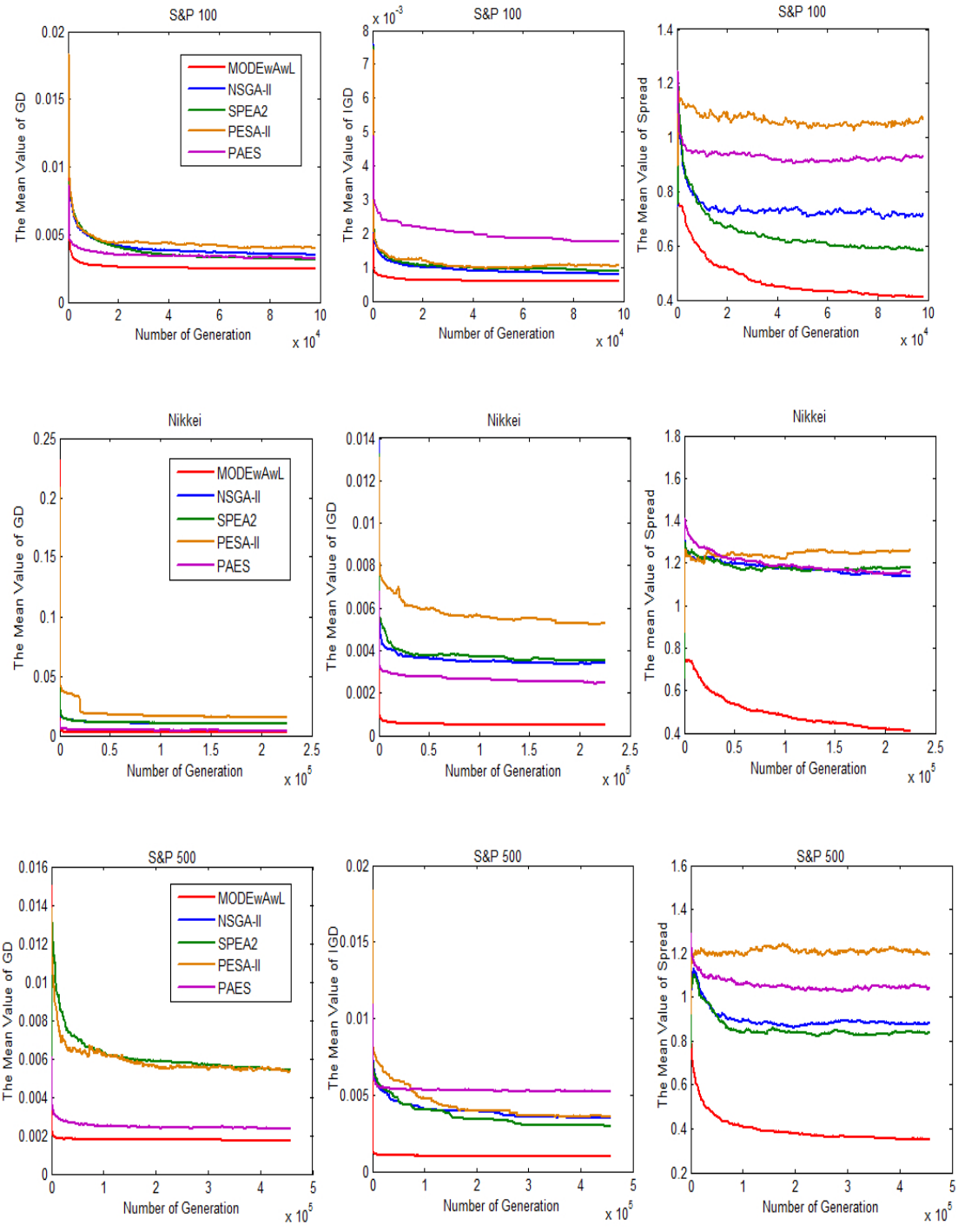


Figure 6.11: Comparisons of convergence of five algorithms.

To gain an intuitive view of the five algorithms over generations, we plot the GD, IGD and Spread (Δ) metrics over generations on six selected instances in Figure 6.11 where the results are averaged over 20 runs. The results confirm that all algorithms considered are able to converge and MODEwAwL is able to converge the fastest in most problem instances.

Algorithm1 $\leftrightarrow$ Algorithm2	Hang Seng	DAX 100	FTSE 100	S & P 100	Nikkei	S & P 500	Russell 2000
MODEwAwL $\leftrightarrow$ NSGA-II	$\sim$	$+$	$+$	$+$	$+$	$+$	$+$
MODEwAwL $\leftrightarrow$ SPEA2	$-$	$+$	$+$	$+$	$+$	$+$	$+$
MODEwAwL $\leftrightarrow$ PESA-II	$\sim$	$+$	$+$	$+$	$+$	$+$	$+$
MODEwAwL $\leftrightarrow$ PAES	$+$	$+$	$+$	$+$	$+$	$+$	$+$
NSGA-II $\leftrightarrow$ SPEA2	$-$	$+$	$\sim$	$\sim$	$+$	$+$	$\sim$
NSGA-II $\leftrightarrow$ PESA-II	$+$	$+$	$+$	$\sim$	$+$	$+$	$\sim$
NSGA-II $\leftrightarrow$ PAES	$+$	$+$	$+$	$+$	$\sim$	$+$	$-$
SPEA2 $\leftrightarrow$ PESA-II	$+$	$\sim$	$\sim$	$\sim$	$+$	$\sim$	$\sim$
SPEA2 $\leftrightarrow$ PAES	$+$	$+$	$+$	$+$	$-$	$\sim$	$-$
PESA-II $\leftrightarrow$ PAES	$+$	$+$	$+$	$+$	$-$	$\sim$	$-$

Table 6.3: Student's t-test results of different algorithms on seven problem instances with $\mathcal{K} = 10$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z_{30} = 1$ and $\vartheta_i = 0.008$.

As stated in Section 3.4.2, IGD can provide the overall performance of an algorithm, measuring its convergence and diversity simultaneously. We compare the IGD values of the five algorithms by using Student's t-test (Walpole et al., 1998). The statistical results obtained by a two-tailed t-test with 38 degrees of freedom at a 0.05 level of significance are given in Table [6.3 $-$ 6.5]. The result of Algorithm1 $\leftrightarrow$ Algorithm2 is shown as "$+$", "$-$", or "$\sim$" when Algorithm1 is significantly better than, significantly worse than, or statistically equivalent to Algorithm2, respectively. Results show that MODEwAwL outperforms other algorithms in most of the problem instances except Hang Seng dataset. For Hang Seng test problem, the performance of SPEA2 outperforms MODEwAwL when $\mathcal{K} = 10$. We therefore can conclude that the proposed MODEwAwL has the

best optimization performance for the portfolio optimization problem with the considered constraints.

Algorithm1 ↔ Algorithm2	Hang Seng	DAX 100	FTSE 100	S & P 100	Nikkei	S & P 500	Russell 2000
MODEwAwL ↔ NSGA-II	∼	+	+	+	+	+	+
MODEwAwL ↔ SPEA2	∼	+	+	+	+	+	+
MODEwAwL ↔ PESA-II	+	+	+	+	+	+	+
MODEwAwL ↔ PAES	+	+	+	+	+	+	+
NSGA-II ↔ SPEA2	+	∼	+	+	+	∼	∼
NSGA-II ↔ PESA-II	+	+	+	+	+	∼	∼
NSGA-II ↔ PAES	+	+	+	+	+	+	∼
SPEA2 ↔ PESA-II	+	∼	∼	∼	+	∼	∼
SPEA2 ↔ PAES	+	+	+	+	−	∼	∼
PESA-II ↔ PAES	+	+	+	+	−	∼	∼

Table 6.4: Student's t-test results of different algorithms on 5 problem instances with $\mathcal{K} = 15$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z_{30} = 1$ and $\vartheta_i = 0.008$.

Algorithm1 ↔ Algorithm2	Hang Seng	DAX 100	FTSE 100	S & P 100	Nikkei	S & P 500	Russell 2000
MODEwAwL ↔ NSGA-II	+	+	+	+	+	+	+
MODEwAwL ↔ SPEA2	+	+	+	+	+	+	+
MODEwAwL ↔ PESA-II	+	+	+	+	+	+	+
MODEwAwL ↔ PAES	+	+	+	+	+	+	+
NSGA-II ↔ SPEA2	−	+	∼	+	+	∼	∼
NSGA-II ↔ PESA-II	+	+	∼	+	+	∼	∼
NSGA-II ↔ PAES	+	+	+	+	−	−	−
SPEA2 ↔ PESA-II	+	∼	∼	∼	+	∼	∼
SPEA2 ↔ PAES	+	+	∼	∼	−	−	−
PESA-II ↔ PAES	∼	+	∼	+	−	−	−

Table 6.5: Student's t-test results of different algorithms on five problem instances with $\mathcal{K} = 5$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z_{30} = 1$ and $\vartheta_i = 0.008$.

6.6 Summary

This chapter presents a new learning-guided multi-objective evolutionary algorithm (MODEwAwL) for mean variance portfolio optimization problems. Four real-world constraints, cardinality, quantity, pre-assignment and round lot, are considered. MODEwAwL adopts a new approach to extend the generic DEMO scheme proposed by Robič and Filipič (2005). The proposed algorithm adopts elitism by maintaining an archive during the evolution process. A learning mechanism is introduced in order to extract important features from the set of elite solutions. An efficient and effective candidate generation scheme utilizing a learning mechanism, problem specific heuristics and effective direction-based search methods is proposed to guide the search towards the promising regions of the search space.

A large set of simulation experiments have been conducted over a number of problem instances. Results demonstrate that the proposed algorithm is highly efficient in terms of both finding solutions close to the true Pareto-front and good distribution along the Pareto-front. Computational results demonstrate that maintaining a secondary population of solutions in combination with a learning-guided candidate solution generation scheme contributes to better performance over four existing well-known MOEAs, NSGA-II, SPEA2, PEAS-II and PAES. The experimental results not only show that the quality of the generated Pareto set approximations significantly improved, but also that the overall computation time can be reduced.

As to the Pareto set approximation, the proposed solution generation scheme embedding learning mechanism, problem specific heuristics and direction-based search methods play a major role, while the efficiency is mainly because the proposed algorithm is computationally cheap as it only uses a single update at each generation. Unlike other MOEAs, MODEwAwL has very few parameters required to set. Performance wise, the proposed MODEwAwL algorithm is not only capable to deliver high-quality portfolios enriched with additional constraints but also able to efficiently solve a reasonable number of assets up to 1318.

Chapter 7

Mean-VaR Portfolio Optimization: A Non-parametric Approach

"Risk, like beauty, is in the eye of beholder."

Leslie A. Balzer

7.1 Introduction

In the MV model, risk is defined by means of dispersion and it is assumed that returns are normally or elliptically distributed (see Section 2.2.4). However, the distributions of returns are asymmetric and usually have excess kurtosis in practice (Cont, 2001; Fama, 1965; Prakash et al., 2003). Variance as a risk measure has thus been widely criticized by practitioners due to its symmetrical measure by equally weighting desirable positive returns against undesirable negative ones. In fact, Markowitz recognized the inefficiencies embedded in the mean-variance approach and suggested the semi-variance risk measure (Markowitz, 1959) in order to measure the variability of returns below the mean. In practice, many rational investors are more concerned with under-performance rather than over-performance in a portfolio. These limitations have led to research directions where realistic risk measures are used to separate undesirable downside movements from desirable upside movements (Biglova et al., 2004). Among those

various risk measures, Value-at-Risk (VaR) (Morgan, 1996) is a popular measurement of risk. Moreover, regardless of the quantification of the risk function, there are usually several practical trading constraints imposed on the solutions of the portfolio optimization problem.

This chapter presents a multi-objective evolutionary algorithm with guided learning for the mean-VaR portfolio optimization problem with practical investment constraints. Six practical trading constraints, namely, cardinality, quantity, preassignment, round lot, class and class limit constraints, are considered. Value-at-Risk (VaR) is used as a risk measure and a nonparametric historical simulation approach is adopted to calculate VaR.

7.2 Value-at-Risk: An Overview

In the literature, researchers and practitioners replace variance by introducing various downside risk measures (Harlow, 1991; Krokhmal et al., 2011) in order to capture realistic market risk exposure by focusing on return dispersions below a specified target. The Safety-First principle introduced by Roy (Roy, 1952) is considered to be fundamental in the development of downside risk measures in the finance literature. Roy's Safety-First criterion suggests selecting a portfolio which minimizes probability of returns falling below some predefined disaster level. Consequently, a growing number of downside risk measures based on lower partial moments have been proposed by academics and practitioners (Bawa, 1975; Bawa and Lindenberg, 1977; Fishburn, 1977; Grootveld and Hallerbach, 1999).

The most popularly embraced technique for measuring downside risk among financial institutions is Value-at-Risk (VaR) (Duffie and Pan, 1997; Jorion, 2006; Linsmeier and Pearson, 2000). VaR measures the maximum likely loss of a portfolio from market risk with a given confidence level (1 - α) over a certain time interval. For instance, if a daily VaR is valued as 100,000 with 95% confidence level, this means that during the next trading day there is only a 5% chance that

the loss will be greater than 100,000. The higher the confidence level, the better the chances that the actual loss will be within the VaR measure. Therefore, the confidence level (1 - α) is usually high, typically 95% or 99%.

There are three main techniques commonly employed to measure VaR: the parametric approach (variance-covariance), nonparametric approach (historical simulation) and Monte Carlo simulation methods (Jorion, 2006; Linsmeier and Pearson, 2000). The choice of the VaR method is critical since the results yielded from each method can be very different from each other (Manganelli and Engle, 2001). Each method has its own strengths and weaknesses. The parametric method assumes financial returns follow a normal or known distribution function whereas the nonparametric (historical simulation) method makes no assumption regarding the distribution. The third method simulates a large number of random scenarios which can be computationally challenging. The analysis conducted by Pérignon and Smith (2010) shows that the most commonly used approach for computing VaR among investment firms that disclose their methodology is historical simulation.

Due to its conceptual simplicity, VaR has been widely recognized by financial regulators and investment practitioners. The Basel Committee for Banking Supervision of the Bank of International Settlements allows financial institutions to use VaR models to set aside regulatory capital amounts that would cover potential operational loss [6] (Jackson et al., 1997). In addition, the Securities and Exchange Commission (SEC) requires financial service firms to provide quantitative information about market risk using the VaR measure (Alexander and Baptista, 2002).

Despite its wide use, VaR has limitations. VaR is widely criticized for not being a coherent risk measure since it does not fulfil a subadditive property. Artzner et al. (1999) show that VaR fails to satisfy the subadditivity property for some distributions of asset returns. In other words, the VaR of a portfolio with two

[6] see Regulation S–K, Item 305, available online at www.sec.gov/divisions/corpfin/forms/regsk.htm

securities may be greater than the combination of VaRs of each security in the portfolio (Acerbi et al., 2001). When VaR is used as the objective function it leads to a non-convex and non-differential risk-return portfolio optimization problem with the number of local optima increasing exponentially with the number of assets (Daníelsson et al., 2008; Gaivoronski and Pflug, 2005; Kolm et al., 2014). In fact, Benati and Rizzi (2007) show that optimization of the mean-VaR portfolio problem leads to a non-convex NP-hard problem which is computationally intractable. Moreover, the non-convexity of VaR discourages diversification.

Practitioners are greatly attracted to the expected shortfall (ES) as an alternative risk measure which considers losses beyond the VaR level (Acerbi et al., 2001; Acerbi and Tasche, 2002). Rockafellar and Uryasev (2002) introduce the expected shortfall under the notion of Conditional Value at Risk (CVaR). Acerbi and Tasche (2002) and Rockafellar and Uryasev (2002) discuss the detailed properties of the expected shortfall (ES). Many studies have applied alternative subadditive risk measures such as Conditional Value at Risk (CVaR) (Rockafellar and Uryasev, 2000) and Partitioned Value-at-Risk (PVaR) (Goh et al., 2012) with corresponding operational consequences.

7.3 Related Work

Given that VaR is the predominantly used quantile-based, industry-standard risk measure, there is a need for efficient algorithms that minimize VaR while obtaining maximum return. In the literature, there are different approaches to measure VaR to investigate portfolio optimization (Charpentier and Oulidi, 2009; Ghaoui et al., 2003; Goh et al., 2012; Natarajan et al., 2008). Although there has been considerable work related to portfolio optimization with various risk measures (Chang et al., 2009; Kolm et al., 2014; Krokhmal et al., 2011), it is noticeable that the number of studies of non-parametric historical VaR in the context of mean-VaR remains relatively small.

Krink and Paterlini (2011) presented a new multi-objective evolutionary algo-

rithm for portfolio optimization, called DEMPO, by hybridizing DE and NSGA-II. The main feature of the DEMPO is the crossover operation of the NSGA-II is replaced by DE operators. Their work considered three different models with variants of risk measures: variance, expected return and value-at-risk. The performance of DEMPO was compared against QP and NSGA-II. The results showed that DEMPO performed better than NSGA-II and comparable to QP. However, their work did not cosider any practical constraints. Gilli and Këllezi (2002) and Gilli et al. (2006) proposed a threshold accepting method to maximize a portfolio's return under VaR and expected shortfall constraints. Dallagnol et al. (2009) employed a genetic algorithm (GA) and particle swarm optimization (PSO) for a mean-VaR portfolio selection problem using historical simulation calculation. Alfaro-Cid et al. (2011) conducted a comparison between mean-variance and mean-VaR approach using multi-objective genetic algorithm. However all these studies have often simplified the problem where practical constraints are not taken into account.

Baixauli-Soler et al. (2011) presented a multi-objective GA for the mean-VaR portfolio optimization problem with minimum transaction units and transaction costs. Computational analysis was performed using fifty assets in Eurostoxx 50 index for 1302 daily historical observations per asset. Results showed the adequacy of the multi-objective GA for solving mean-VaR problem. Jevne et al. (2012) also studied the mean-VaR portfolio optimization problem with minimum transaction units and transaction costs and investigate the effect of the initialization scheme on the results with multi-objective differential evolution and NSGA-II. Experimental results showed that the refined initialization scheme improves the convergence of both algorithms. Computational analysis was performed using 100 assets in the S&P 100 index for 250 daily historical observations.

Anagnostopoulos and Mamanis (2011a) replaced the variance risk measure with VaR and expected shortfall (ES). Three multi-objective evolutionary algorithms, SPEA2, NSGA-II and PESA, were compared against exact methods to evaluate the portfolio selection problem with cardinality, quantity and class constraints. Computational results were conducted using 96 assets in S&P 100 index for daily

return from 3 March 2004 to 20 February 2008. Results showed that NSGA-II and SPEA2 are competitive to each other and they outperformed PESA in terms of hypervolume measure.

7.4 Problem Model

In this work, six practical trading constraints, cardinality, quantity, pre-assignment, round lot, class and class limit, are considered (see Section 2.4). The mean-VaR model (see Section 2.3.2) can be extended with these constraints as follows:

$$min \qquad \psi(w) \tag{7.1}$$

$$max \qquad \mu(w) \tag{7.2}$$

$$subject\ to \quad \sum_{i=1}^{\mathcal{N}} w_i \leq 1, \quad 0 \leq w_i \leq 1 \tag{7.3}$$

$$\sum_{i=1}^{\mathcal{N}} s_i = \mathcal{K}, \tag{7.4}$$

$$w_i = y_i.\vartheta_i, \quad i = 1, ..., \mathcal{N}, \quad y_i \in \mathbb{Z}_+ \tag{7.5}$$

$$\epsilon_i s_i \leq w_i \leq \delta_i s_i, \quad 0 \leq \epsilon_i \leq \delta_i \leq 1, \quad i = 1, ..., \mathcal{N} \tag{7.6}$$

$$L_m \ \leq \sum_{s_i \ \in \ C_m} w_i \ \leq U_m, \quad m = 1, ...\mathcal{M}, \tag{7.7}$$

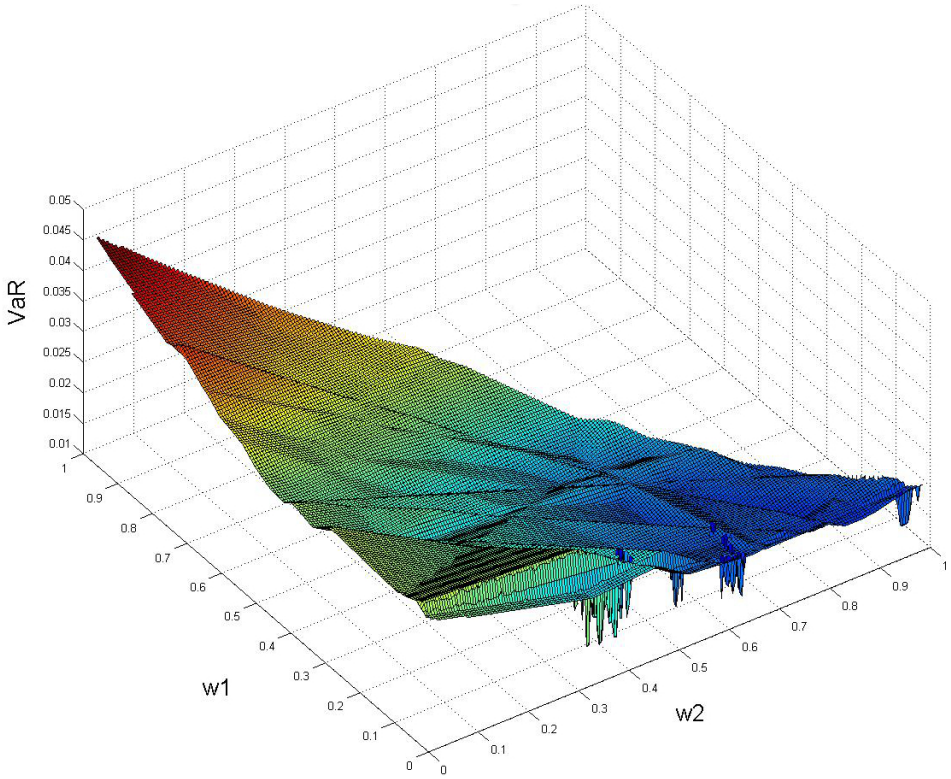
$$s_i \geq z_i, \quad i = 1, ..., \mathcal{N} \tag{7.8}$$

$$s_i, z_i \in \{0, 1\}, \quad \{z_i \in Z \mid z_i == 1\}, \quad i = 1, ..., \mathcal{N} \tag{7.9}$$

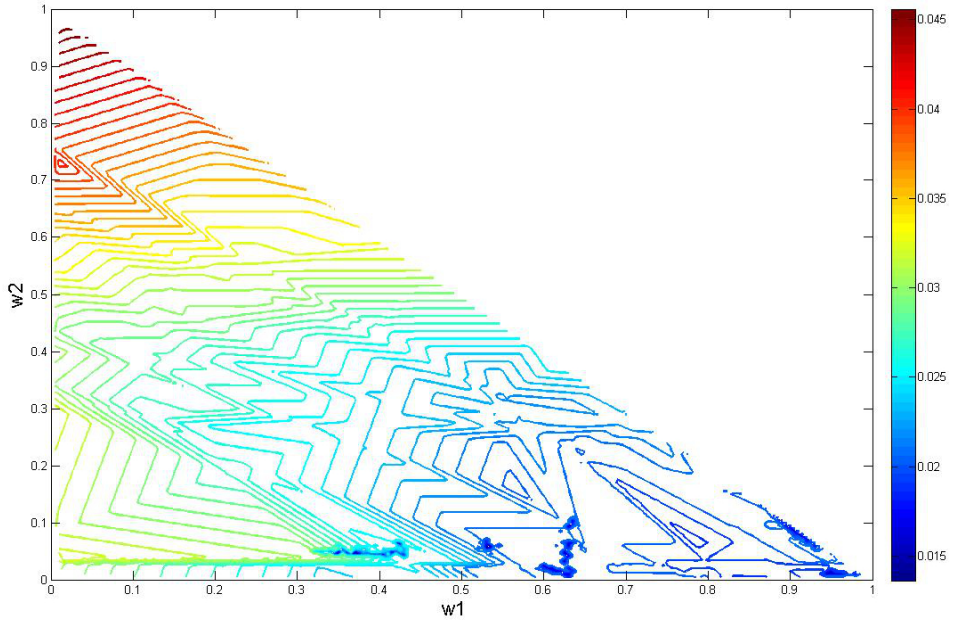
where Eq. (7.1) and Eq. (7.2) are risk and return objectives respectively (see Section 2.3.2). Eq. (7.4) defines the cardinality constraint where $\mathcal{K}$ is the number of invested assets in the portfolio and the binary variable s_i denotes whether asset i is invested or not. Eq. (7.6) defines the quantity constraint. If asset i is invested, the proportion of capital w_i lies in $[\epsilon_i, \delta_i]$. Eq. (7.8) defines the pre-assignment constraint where the binary vector z_i denotes if asset i is in the pre-assigned set Z that has to be included in the portfolio or not. Eq. (7.5) defines the round lot constraint where y_i is a positive integer variable and ϑ_i is the minimum lot that can be purchased for each asset (see Section 2.4.3). Eq. (7.7) defines class and class limit constraints. Every asset is classified in a certain class (i.e., $i \in C_m$) and class $C_m, m = 1, \ldots, \mathcal{M}$, represents $\mathcal{M}$ mutually exclusive sets of assets and L_m and U_m denote the lower and upper proportion limit for class m. In this work, it is assumed that $L_m > 0$ for every class C_m and $\mathcal{K} \geq \mathcal{M}+ \mid Z \mid - \mid Z^{cl} \mid$ where Z^{cl} is the set of classes represented by the preassigned set Z (i.e., $Z^{cl} = \{m : z_i \in C_m\},\ z_i \in Z$). For instance, if $\mathcal{K} < \mathcal{M}$, there exists no feasible solutions satisfying the class constraint.

7.5 MOEA with Guided Learning

Minimizing VaR as a risk measure is a challenging task due to the non-smooth objective function landscape with many local minima. Figure 7.1 shows the surface and contour plots of the Value-at-Risk (VaR) of portfolios in a three assets universe (Coca-Cola Co., 3M Co. and Halliburton Co.) displaying the existence of non-smooth and non-convex surface with several local minima. The results are calculated for basic mean-VaR portfolio optimization by using historical simulation with 3 years of data and 99% confidence interval. The triangular shaped area delimits the feasible solution area. Figure 7.1(a) illustrates the non-convex objective function of VaR for the portfolio optimization problem in a three asset universe. Figure 7.1(b) depicts the existence of several local minima such as those located at $w_1 = 583159$, $w_2 = 0.172896$ with $\text{VaR}_{99} = 0.020634$ and $w_1 = 0.789138$, $w_2 = 0.10543$ with $\text{VaR}_{99} = 0.019671$.



(a) Surface plot



(b) Contour plot

Figure 7.1: The historical VaR of feasible portfolios comprising of three stocks (Coca-Cola Co., 3M Co. and Halliburton Co.) with 3 years of data and 99% confidence interval. w_1 is the proportion of investment in Coca-Cola, w_2 is the proportion of investment in Halliburton. The amount investment in 3M is equal to $1 - w_1 - w_2$. Short selling is not allowed.

In practice, portfolios are composed of markets with potentially hundreds to thousands of available assets, and the calculation of risk measures grows quickly in relation to the number of assets. When more dimensions and trading constraints are added to the problem, the complexity of the problem increases. Optimal exponential algorithms for reasonable problem dimensions are still not available. Approximation approaches such as smoothing (Gaivoronski and Pflug, 2005) and meta-heuristics are the known alternatives to find optimal or near-optimal portfolios in a reasonable amount of time.

This section presents a MOEA with guided learning (MODE-GL) for the constrained mean-VaR portfolio optimization problem. MODE-GL is mainly adapted from MODEwAwL presented in the previous Chapter (see Section 6.4). The main difference of MODE-GL from MODEwAwL is outlined as follows:

- An additional small archive D is introduced to maintain $\mathcal{D}_{size}$ least crowded solutions (see Section 7.5.5).
- Two different variants of the DE mutation schemes in the solution generation scheme are proposed (see Section 7.5.3).

These two amendments are proposed to promote exploration and to surmount valleys in the objective landscape. Like MODEwAwL, MODE-GL maintains an archive A with non-dominated solutions throughout the evolution. A learning mechanism is used to extract important features of non-dominated solutions in the archive A. These features are exploited in generating promising offspring solutions. MODE-GL thus aims to promote convergence by concentrating on the promising regions of the search space. Two proposed variants of differential evolution mutation schemes are utilized in order to promote the exploration of the search towards the least crowded region of the solution space. The pseudocode of the proposed algorithm is described in Algorithm 7.1.

Algorithm 7.1: Procedure of MODE-GL.

Input: $\mathcal{NP}$: the number of individuals in the population P,
$\mathcal{A}_{size}$: the size of the archive A,
$\mathcal{D}_{size}$: the size of the archive D,
$\mathcal{CR}$: the crossover probability,
$\mathcal{F}$: a scaling factor ;

Output: A;

$P := \emptyset$; $A := \emptyset$; $D := \emptyset$;
$P \leftarrow$ randomly create initial population P;
while *stopping criterion not met* **do**
    // Update the archive A and archive D (see Section 7.5.5)
    $A \leftarrow$ *update* archive A with non-dominated solutions from P;
    if $\mid A \mid \geq \mathcal{A}_{size}$ **then**
        $A \leftarrow$ maintain archive A with $\mathcal{A}_{size}$ least crowded solutions;
    $D \leftarrow$ *update* archive D with $\mathcal{D}_{size}$ least crowded solutions from P;
    // Learning mechanism (see Section 6.4.3)
    learn from the archive A to identify the promising asset(s);
    for *each individual* p_i *(*$i = 1, \ldots, \mathcal{NP}$*)* **do**
        // Candidate generation (see Section 7.5.3)
        $p' \leftarrow$ create new candidate p' from P and learning mechanism;
        if *candidate* p' *does not satisfy constraints* **then**
            repair p' (see Section 7.5.4);
        evaluate the candidate p' by f_1 and f_2 (see Eq. 7.1, 7.2);
        if p' *dominates* p_i **then**
            p' replaces p_i;
        else if p_i *dominates* p' **then**
            discard p';
        else
            add p' to the current population P;
    if $\mid P \mid \geq \mathcal{NP}$ **then**
        // Truncate P (see Section 6.4.7)
        $P \leftarrow$ maintain P with the best $\mathcal{NP}$ solutions, ranked by
            non-domination and crowding distance metrics;
    randomly enumerate the individuals in P;

7.5.1 Solution Representation and Encoding

Assume an array Γ consists of $\mathcal{M}$ real values each one representing the total proportion invested in each class; an array S contains $\mathcal{K}$ integer numbers, each representing a selected asset in the portfolio; and an array W includes $\mathcal{K}$ real values, representing the allocation of each selected asset in the portfolio. We present the following representation scheme to handle several considered constraints:

$$
\begin{aligned}
\Gamma =& \{\theta_1, \ldots, \theta_{\mathcal{M}}\}, \;\; 0 \leq \theta_m \leq 1, \;\; m = 1, \ldots, \mathcal{M}, \\
S =& \{s_1, \ldots, s_{\mathcal{M}}, \ldots, s_Q, s_{Q+1}, \ldots, s_{\mathcal{K}}\}, \\
& Q = \mathcal{M}+ \mid Z \mid - \mid Z^{cl} \mid, \\
& s_b \in C + Z, \;\; s_j \in \{1, \ldots, \mathcal{N}\} - \{s_1, \ldots, s_Q\}, \\
& b = 1, \ldots, Q, \;\; j = Q+1, \ldots, \mathcal{K}, \\
W =& \{w_i, \ldots, w_{\mathcal{K}}\}, \;\; 0 \leq w_i \leq 1, \;\; i = 1, \ldots, \mathcal{K}.
\end{aligned}
$$

With this solution representation, the cardinality constraint is satisfied by the array S which has specified size $\mathcal{K}$. The preassignment constraint is satisfied by including all preassigned assets in S. The set of classes represented by the preassigned assets in Z are identified and denoted by Z^{cl}. We then ensure that $\mathcal{M}- \mid Z^{cl} \mid$ assets are selected from each remaining class. As stated in Section-7.4, in this work, it is assumed that $\mathcal{K} \geq \mathcal{M}+ \mid Z \mid - \mid Z^{cl} \mid$. If $\mathcal{K} > Q$, then the remaining $\mathcal{K} - Q$ assets are randomly selected from the available unselected assets. In the literature, Anagnostopoulos and Mamanis (Anagnostopoulos and Mamanis, 2011a) use a similar encoding scheme but their model does not consider either pre-assignment or round lot constraints.

Given that $\mathcal{N} = 94$ and $\mathcal{M} = 6$ where $C_1 \in \{1, \ldots, 15\}$, $C_2 \in \{16, \ldots, 30\}$, $C_3 \in \{31, \ldots, 45\}$, $C_4 \in \{46, \ldots, 60\}$, $C_5 \in \{61, \ldots, 75\}$ and $C_6 \in \{76, \ldots, 94\}$, an

example portfolio with $\mathcal{K} = 10$ would be represented as described below:

$$
\begin{aligned}
Z &= \{30\}, \quad Z^{cl} = \{C_2\}, \quad \mid Z^{cl} \mid = 1, \\
\Gamma &= \{0.05, 0.05, 0.05, 0.05, 0.05, 0.05\}, \\
S &= \{8, 17, 30, 47, 62, 85, 31, 92, 37, 69\}, \\
W &= \{w_8 = 0.112, w_{17} = 0.048, w_{30} = 0.024, w_{47} = 0.376, w_{62} = 0.024, \\
&\quad w_{85} = 0.136, w_{31} = 0.12, w_{92} = 0.064, w_{37} = 0.064, w_{69} = 0.032\}.
\end{aligned}
$$

7.5.2 Initial Population Generation

To generate an initial population, all assets in the pre-assignment set Z are included first and the remaining $\mathcal{K} - \mid Z \mid$ are randomly selected by making sure at least one asset from each class of M is included. The proportions (with exact lots) are assigned to those $\mathcal{K}$ selected assets randomly. If the generated portfolio violates the budget, quantity and/or class limit constraints, such a solution is corrected by the constraint handling techniques detailed in Section 7.5.4. This ensures that all generated solutions in the population are feasible.

7.5.3 Candidate Generation

One of the factors to consider in designing the portfolio model in MODE-GL is to find an effective way to generate offspring. In this section, an effective and efficient candidate generation scheme with a good balance between exploitation and exploration is proposed. A new solution is generated in two phases: the selection of assets from a universe of $\mathcal{N}$ available assets and the allocation of capital to those selected assets. In the first phase, the learning mechanism (see Section 6.4.3) together with problem specific heuristics (**S1** - **S4**) described below are exploited to identify promising assets while directing the search towards the most promising regions of the search space.

In order to generate a new candidate solution, the $\mid Z \mid$ pre-assigned assets are first selected. By taking into account the above stated intuitive learning, in this

work, MODE-GL then randomly uses the following selection schemes until the remaining assets ($\mathcal{K} - \mid Z \mid$) have been selected, while making sure at least one asset from each class of $\mathcal{M}$ is included. By adopting the selection schemes stated below, it is ensured that the new candidate solution satisfies the pre-assignment, class and cardinality constraints.

S1: Roulette wheel selection based on the concentration score c_i.

S2: Select asset with the highest concentration score c_i.

S3: Select asset with the highest mean return values.

S4: Select asset with the least standard deviation of return values.

In the second phase, the proportions of those selected assets for the new candidate solution are assigned by using two extended variants of DE mutation schemes as follows:

W1: $w_i' := best_i + r[0,1] \times (w1_i - w2_i)$

W2: $w_i' := w_i + \mathcal{F} \times (best_i - w_i) + \mathcal{F} \times (w1_i - w2_i)$

where $\mathcal{F}$ is the scaling factor for differential evolution. The two portfolios ($p1 \in D$ and $p2 \in D$ with $w1_i$ and $w2_i$ allocations respectively) are randomly selected from the least crowded portfolio archive D and $best$ is the best solution randomly selected from the best 10% of archive A.

These two DE mutation schemes are extended from similar variants of *DE/best/1* (Das and Suganthan, 2011) and *DE/current-to-pbest/1* (Zhang and Sanderson, 2009). In our extended version the difference is that $w1_i$ and $w2_i$ are randomly selected portfolios from archive D to direct the search towards promising unexplored directions. The detailed procedure of the candidate generation is provided in Algorithm 7.2. The proposed candidate generation mechanism intends to guide the search towards promising directions by learning from the best found solutions from the archive A. In this way, the proposed algorithm converges ef-

ficiently. The new candidate portfolio is repaired if the quantity and round lot constraints are violated (see the repair mechanism in Section 7.5.4).

Algorithm 7.2: Procedure of generating a candidate solution.

Input: concentration score of assets $c_i(i = 1, ..., \mathcal{N})$ and $\bar{p} \in P$;
Output: candidate solution p';

```
select | Z | from preassigned set Z and randomly select K− | Z | assets by S1, S2, S3 and/or S4 while ensuring class constraint;
randomly select three different portfolios: p1, p2, p3 ∈ {P \ p̄};
randomly select an index i from those K assets and assign i to j and γ;
for each selected asset do
    if r(0,1) < CR || j == γ then
        allocate weight w' by W1 or W2;
    else
        assign weight w' with corresponding w̄ of parent portfolio p̄;
    randomly select an index i from those K selected and assign i to j;
```

7.5.4 Constraint Handling

When using an evolutionary algorithm to solve constrained optimization problems, various methods have been proposed in the literature for handling constraints, such as penalty function methods, special representation and operator methods, repair methods, separation of objective and constraint methods, and hybrid methods (Coello, 2002). Among those methods, the repair method is one of the commonly adopted approaches to locate feasible solutions for combinatorial optimization problems (Coello, 2002; Salcedo-Sanz, 2009).

During the population sampling, each constructed individual portfolio is repaired if it does not satisfy all considered constraints. As described in Section 7.5.3, the new solution generated by MODE-GL already satisfies the cardinality, class and

pre-assignment constraints. Hence, the following repair mechanism is applied:

1. All weights of the selected asset in the candidate solution are adjusted by setting:
$$w_i' = \psi_i + \frac{w_i' - \psi_i}{\sum(w_i' - \psi_i)}$$
where the smallest divisible lot $\psi_i = inf\{x_i : x_i \,\%\, \vartheta_i == 0 \ and \ x_i \geq \epsilon_i\}$.

2. The weights are then adjusted to the nearest round lot level by setting $w_i' = w_i' - (w_i' \ mod \ \vartheta_i)$. The remaining amount of capital is redistributed in such a way that the largest amount of $(w_i' \ mod \ \vartheta_i)$ is added in multiples of ϑ_i until all the capital is spent.

3. The weights are then adjusted (if the class limit constraints are violated). If $\theta_m < L_m$, insert $y := L_m - \theta_m$ to the underflow class and subtract ϑ_i from those classes where $L_{m'} + \vartheta_i \leq \theta_{m'} \leq U_{m'}$ until $\sum \vartheta_i \geq y$. Similarly, the same for the overflow class. This process is repeated until all limits are satisfied.

7.5.5 Maintaining Archives

The main objective of the external archive A is to maintain the well-spread non-dominated solutions encountered during the search. In each generation, archive A is updated with the non-dominated solutions from the trial population. The computational time to maintain the archive increases with the archive size (Coello et al., 2004; Knowles and Corne, 2000; Zitzler et al., 2001). The size of the archive is therefore restricted to a pre-specified value. When the external archive has reached its maximum capacity $\mathcal{A}_{size}$, the most crowded non-dominated members are identified and discarded.

In addition, in each generation, a small number of the least crowded solutions are maintained in archive D and they are not required to be efficient. As noted, mean-VaR objective function landscapes are inclined to have many local minima (see Figure 7.1) and therefore the search needs to cover sufficient solution space to maximize the probability of discovering the global optimum. The least

crowded solutions from archive D are exploited to promote the exploration of the search towards the least explored region of the solution space in order to achieve well-spread efficient solutions.

7.6 Performance Evaluation

In order to evaluate the overall performance of MODE-GL, we compare it with two well-known multi-objective evolutionary algorithms in the literature, namely NSGA-II and SPEA2. Moreover, learning mechanism has been incorporated into NSGA-II and SPEA2 in order to investigate the impact of the mechanism. The four algorithms compared with MODE-GL are as follows:

- NSGA-II: the Non-dominated Sorting Genetic Algorithm II proposed by Deb et al. (2002). The algorithm uses binary tournament selection based on the crowding distance. It performs crossover and mutation by simulated binary crossover and polynomial mutation operators (see Section 3.3.2.1).
- SPEA2: the Strength Pareto Evolutionary Algorithm proposed by Zitzler et al. (2001). The algorithm employs fine-grained fitness assignment, density estimation techniques and archive truncation methods. Like NSGA-II, it uses binary tournament selection, simulated binary crossover and polynomial mutation evolutionary operators (see Section 3.3.2.2).
- NSGA-II-GL: Learning mechanism is incorporated into the binary crossover scheme of NSGA-II.
- SPEA2-GL: Learning mechanism is incorporated into the binary crossover scheme of SPEA2.

To conduct a fair comparison, we use the same population size and archive size (if applicable) for all the algorithms tested in this work. We have chosen to run all the algorithms with the same stopping criteria (i.e. the same number of evaluations) to generate the Pareto front. All algorithms considered in this study were coded in C# using the information from the relevant papers and run on

an Intel Corei7 with 3.20GHz and 2.79GHz processors and 6GB RAM. Each algorithm also uses the same encodings (see Section 7.5.1) and repair mechanism (see Section 7.5.4) when a newly constructed portfolio violates the considered constraints. Before the experiments were performed, parameters were tuned for all algorithms using DS1. Table 7.1 shows the best parameter settings used for each of the algorithms.

Parameters	MODE-GL	NSGA-II	SPEA2	NSGA-II-GL	SPEA2-GL
Number of Population ($\mathcal{NP}$)	100	100	100	100	100
Number of Generation	$5,000\mathcal{N}$	$5,000\mathcal{N}$	$5,000\mathcal{N}$	$5,000\mathcal{N}$	$5,000\mathcal{N}$
Scaling Factor ($\mathcal{F}$)	0.3	-	-	-	-
Crossover Probability ($\mathcal{CR}$)	0.9	0.9	0.9	0.9	0.9
Crossover Distribution Index	-	20	20	20	20
Mutation Probability	-	$1/\mathcal{N}$	$1/\mathcal{N}$	$1/\mathcal{N}$	$1/\mathcal{N}$
Mutation Distribution Index	-	20	20	20	20
Tournament Round	-	-	1	-	1
$\mathcal{A}_{size}$	100	-	100	-	100
$\mathcal{D}_{size}$	10	-	-	-	-

Table 7.1: Parameter Setting of the Algorithms.

In this section, we perform a set of experiments using two datasets DS1 and DS2 (see Section 2.5) to investigate the potential of MODE-GL for multi-objective constrained portfolio optimization problems and compare it with four other algorithms, NSGA-II, SPEA2, NSGA-II-GL and SPEA2-GL. Two performance measures, IGD and HV, are used to evaluate the performance of the tested algorithms. Experimental results obtained for each algorithm are the average of 30 runs.

The results of IGD, HV and running time of the five algorithms performed on first dataset (DS1) are shown in Figure 7.2. These results are obtained for the constrained portfolio optimization problem with cardinality $\mathcal{K} = 10$, floor $\epsilon_i = 0.01$ and ceiling $\delta_i = 1.0$, pre-assignment $Z = \{30\}$, round lot $\vartheta_i = 0.008$,

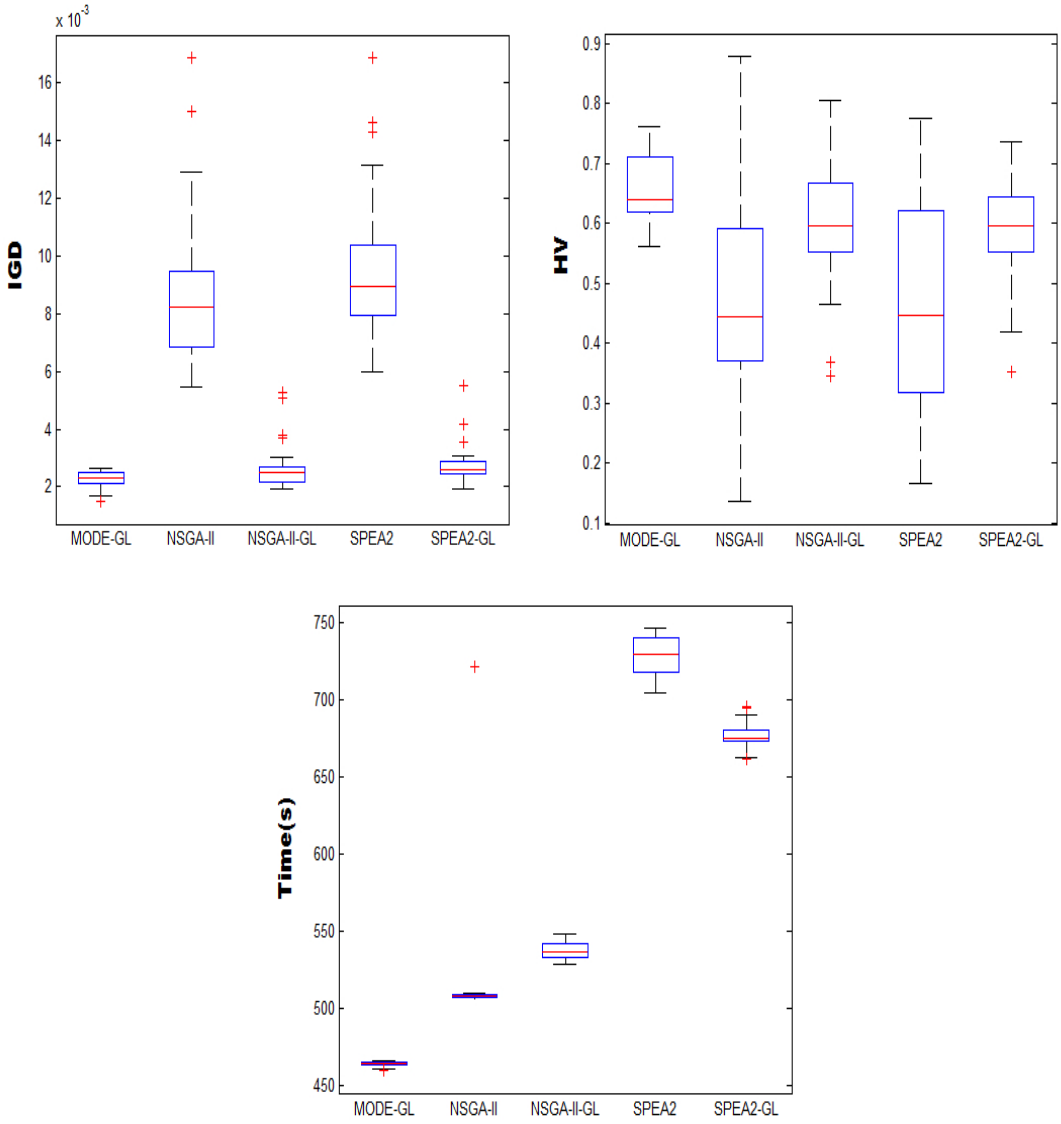


Figure 7.2: Performance of algorithms in terms of IGD, HV and computational time for S & P 100.

class $\mathcal{M} = 6$ with $15, 15, 15, 15, 15, 19$ assets in each class (i.e., $C_1 \in \{1, \ldots, 15\}$, $C_2 \in \{16, \ldots, 30\}$, $C_3 \in \{31, \ldots, 45\}$, $C_4 \in \{46, \ldots, 60\}, C_5 \in \{61, \ldots, 75\}, C_6 \in \{76, \ldots, 94\}$ and $L_m = 0.05$ for each $m = 1, \ldots, 6$. Given that the lower bound of 5% as the class limit specifies an upper bound of 75% of investment in each

class/category, no upper limits have been specified.

The results show that the proposed MODE-GL obtains the smallest mean value for inverted generational distance (IGD) and the largest mean value for hypervolume (HV), compared with the other four algorithms, demonstrating the best performance among the five algorithms. NSGA-II and SPEA2 have similar performance and both have slow convergence compared to MODE-GL. SPEA2 and SPEA2-GL are the most computationally expensive algorithms in terms of CPU time. When the learning-guided solution generation mechanism is incorporated into NSGA-II and SPEA2, the performance of the algorithms improves significantly. Therefore, we would conclude that the learning-guided solution generation mechanism promotes the effective convergence of the search.

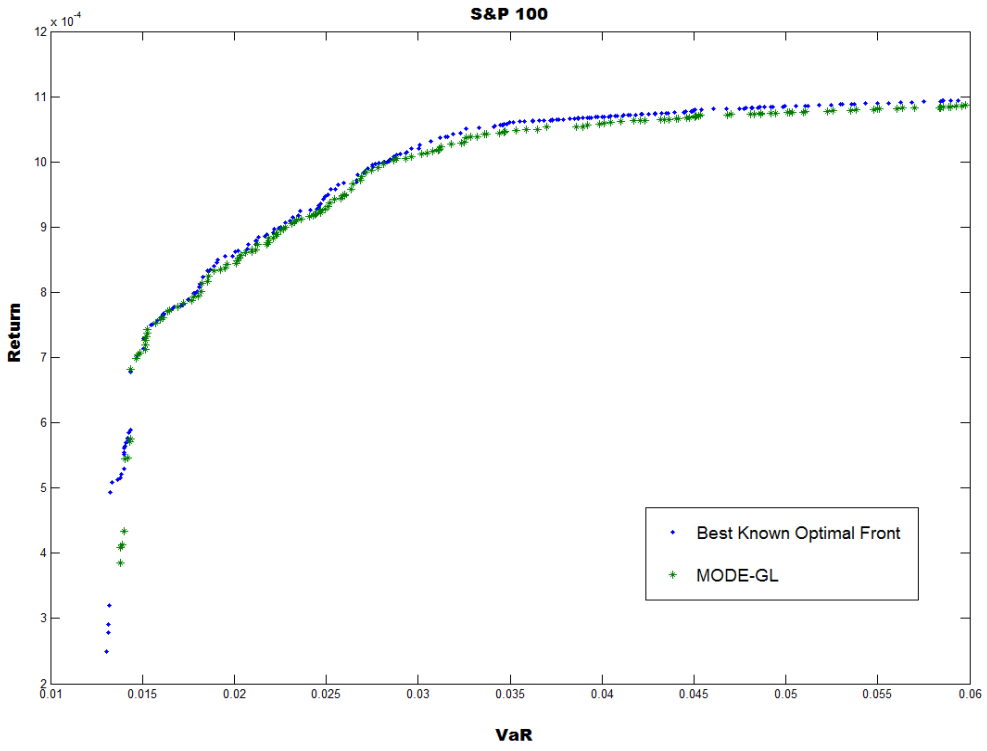


Figure 7.3: S & P 100: Comparison of obtained efficient frontiers of each algorithm together with the best known optimal front obtained from all tested algorithms.

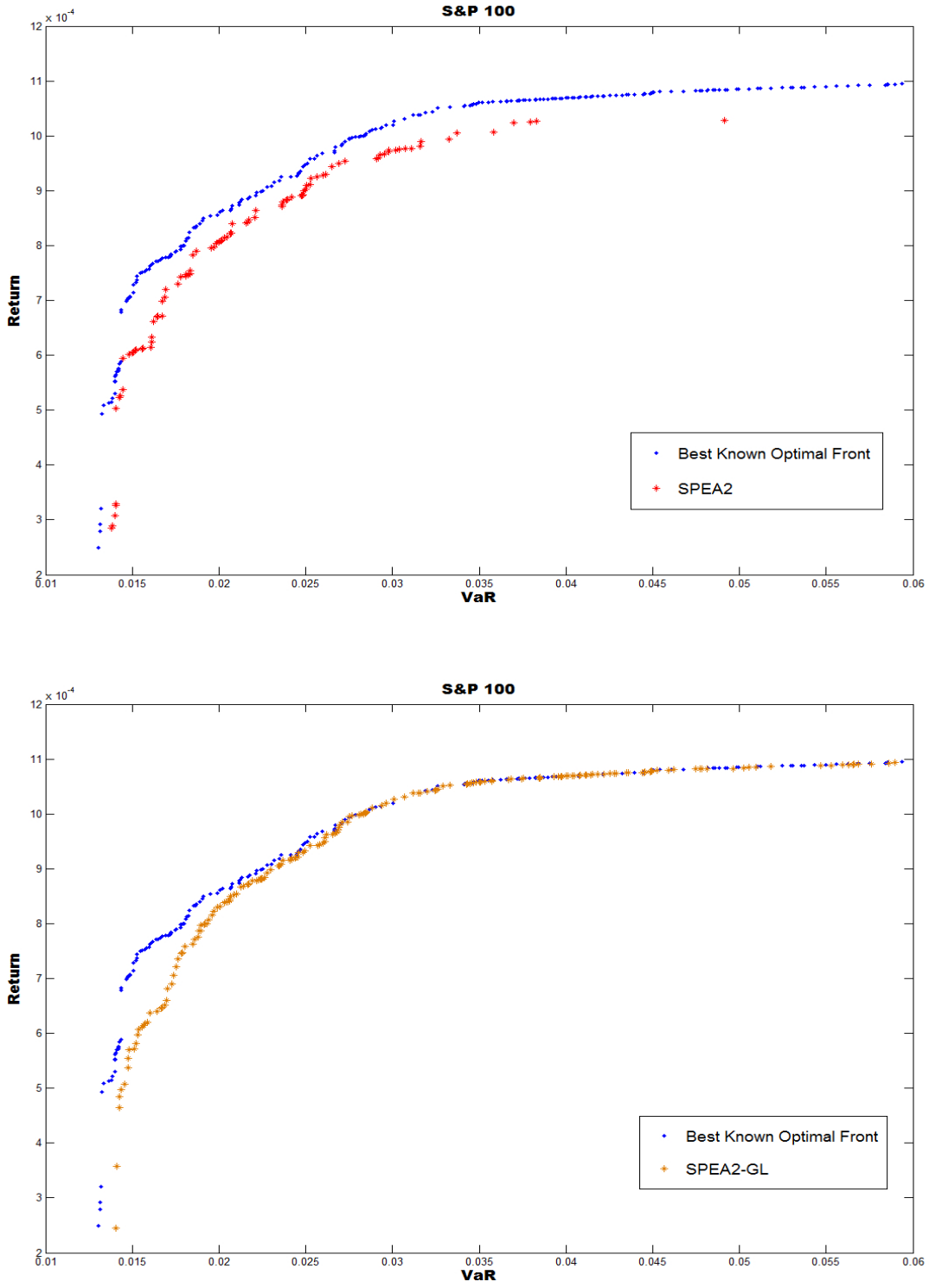


Figure 7.3: S & P 100: Comparison of obtained efficient frontiers of each algorithm together with the best known optimal front obtained from all tested algorithms.

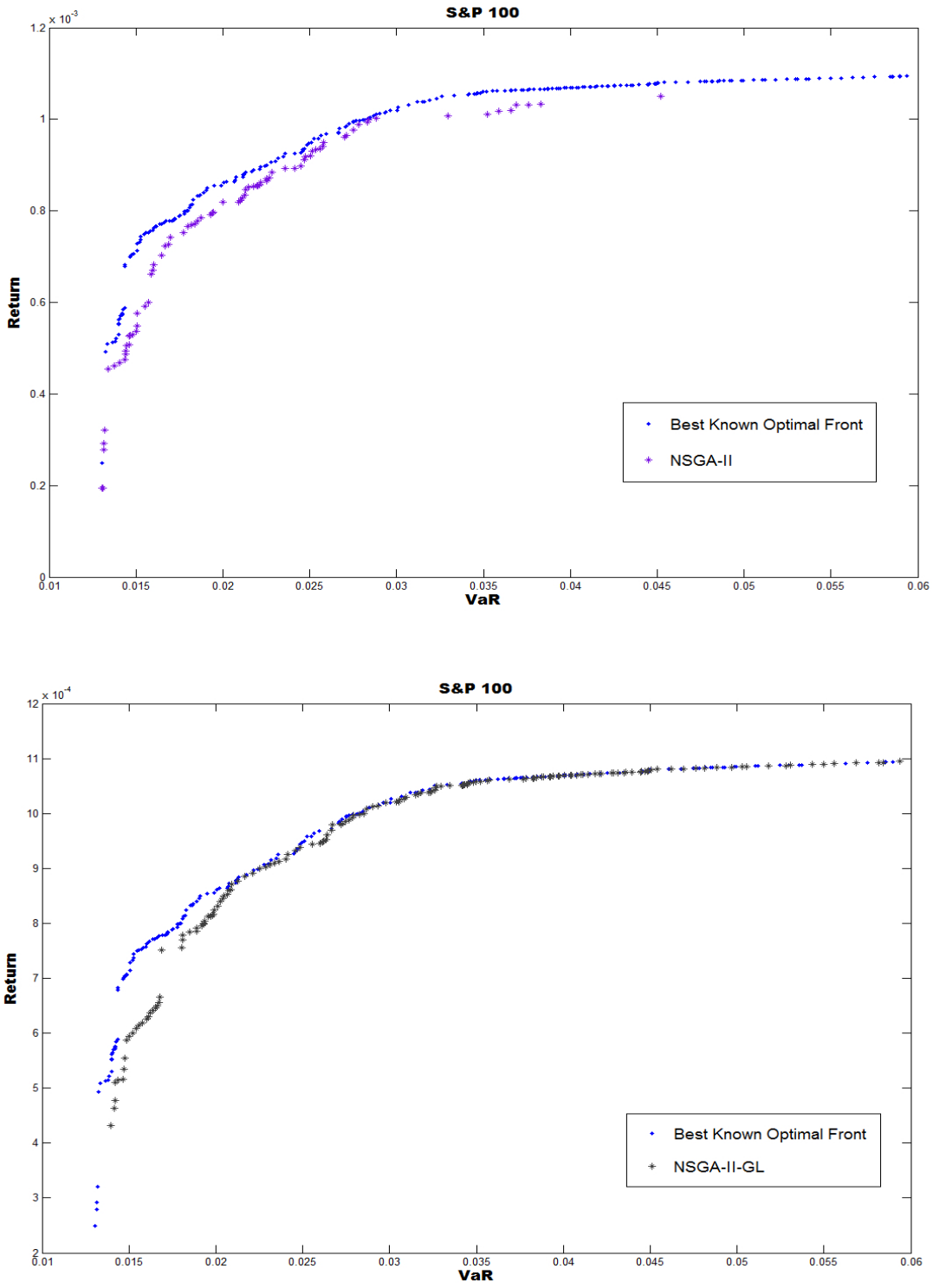


Figure 7.3: S & P 100: Comparison of obtained efficient frontiers of each algorithm together with the best known optimal front obtained from all tested algorithms.

As noted in Section 3.2.3, the optimal efficient frontier of the constrained portfolio optimization is not known for the tested datasets. The best known efficient frontier is thus obtained by collecting all the non-dominated portfolios produced from all the tested algorithms. For illustrative purposes, the obtained efficient frontiers of the tested algorithms for DS1, compared with the best known estimated efficient frontier, are provided in Figure 7.3. The horizontal axis describes the loss that might be incurred with a probability $\alpha = 0.01$. Figure 7.3 shows that MODE-GL, NSGA-II-GL and SPEA2-GL provide a very good approximation of the efficient frontier. The performance of both NSGA-II and SPEA2 improves significantly when the learning-guided solution generation scheme is incorporated.

Figure 7.4 shows how the composition of the securities varies over the range of portfolio risk for the dataset DS1. The results are generated from efficient solutions obtained from a single run of each algorithm and it shows that allocation to all asset classes is present and the preassigned constraint is also satisfied.

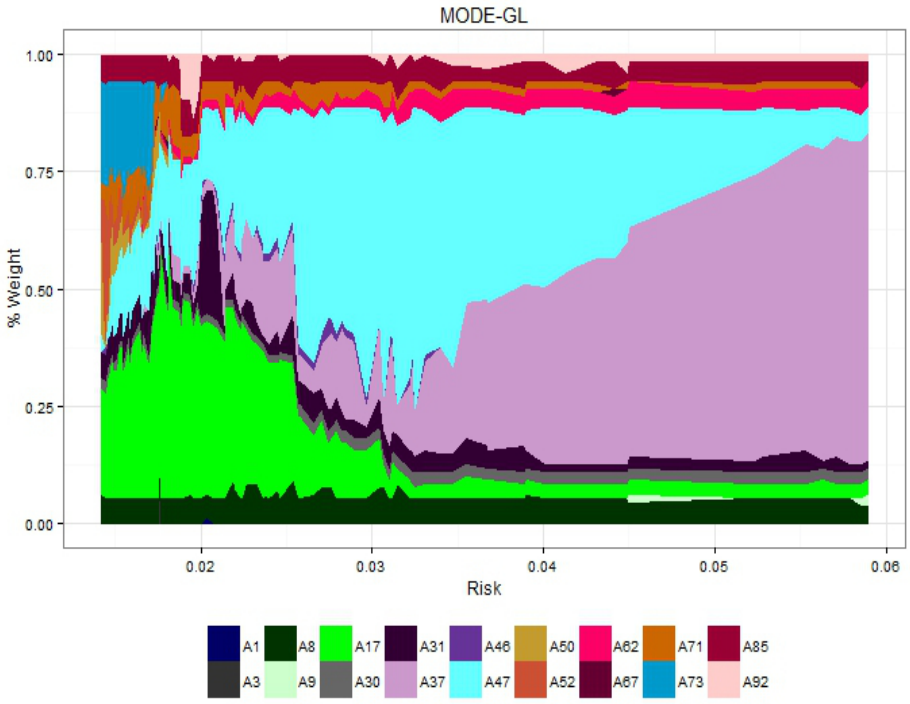


Figure 7.4: S & P 100: Transaction map for portfolio risk.

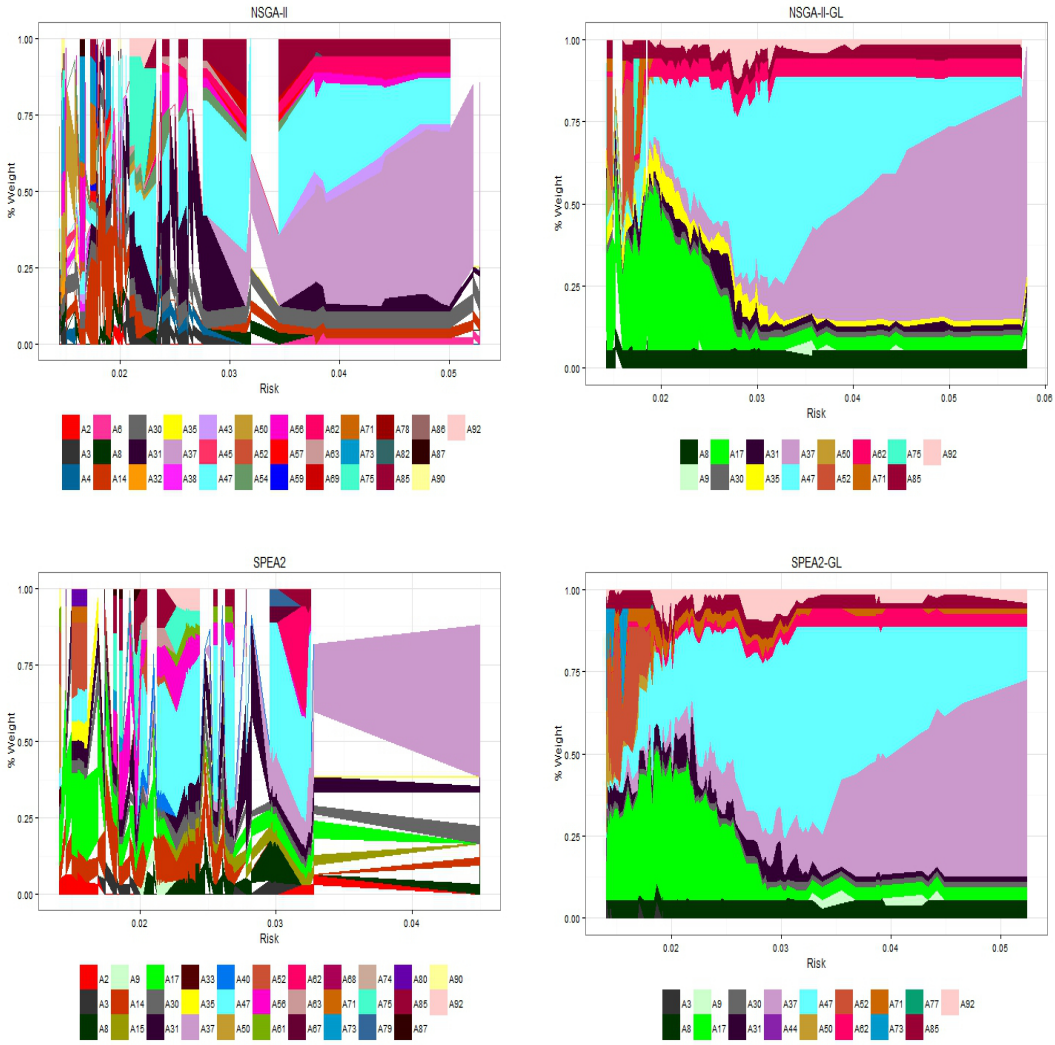


Figure 7.4: S & P 100: Transaction map for portfolio risk.

In each case, the figure depicts how the obtained portfolio is allocated for an obtained level of risk. Each color represents one of the asset selected in the obtained pareto set. A vertical strip through the bands (without white space) indicates the obtained portfolio allocations at that risk level. A vertical strip through the bands (with white space) indicates that no feasible solution can be found for a specific risk level. This discontinuity can also be seen in the obtained efficient frontier as depicted in Figure 7.3. When the learning mechanism is adopted, the

obtained results indicate that the composition of the assets transitions smoothly from one risk level to another.

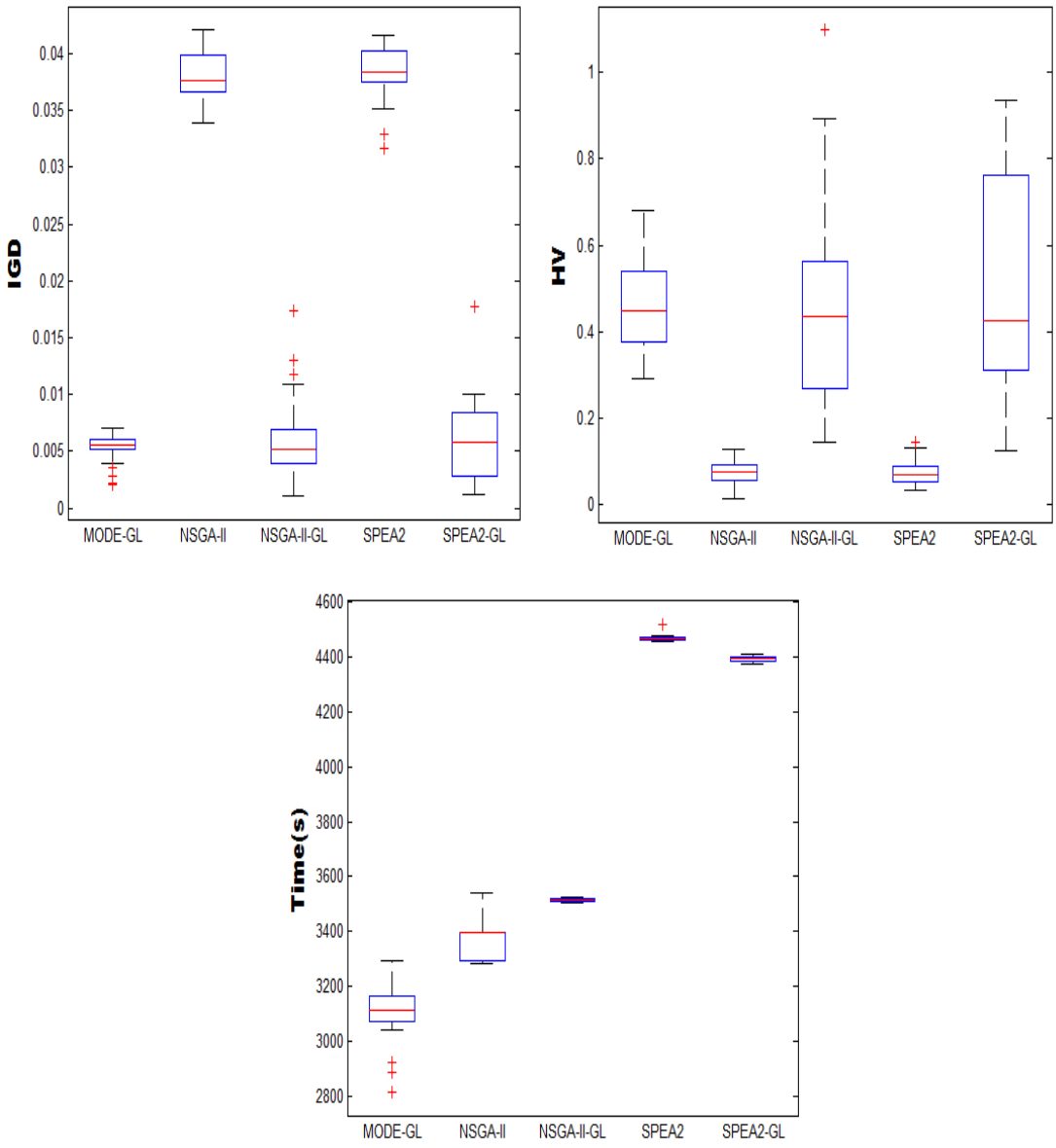


Figure 7.5: Performance of algorithms in terms of IGD, HV and computational time for S & P 500.

The results for IGD, HV and running time of the five algorithms performed on the second dataset (DS2) are shown in Figure 7.5. These results are obtained for the constrained portfolio optimization problem with cardinality $\mathcal{K} = 20$, floor $\epsilon_i = 0.01$ and ceiling $\delta_i = 1.0$, pre-assignment $Z = \{30\}$, round lot $\vartheta_i = 0.008$, class $\mathcal{M} = 19$ with 25 assets in each class (i.e., $C_1 \in \{1, \ldots, 25\}, \ldots, C_{19} \in \{451, \ldots, 475\}$) and $L_m = 0.05$ for each $m = 1, \ldots, 19$. Given that the lower bound of 5% as class limit specifies an upper bound of 10% of investment in each class/category, no upper limits have been specified.

The results show that MODE-GL obtains the largest mean value for hypervolume measure and it is very competitive to NSGA-II-GL and SPEA2-GL in terms of IGD measure. The results also show that SPEA2 ranks as the most inefficient both in terms of solution quality and computing time. The obtained results for DS2 reaffirm the effectiveness of the incorporation of the learning mechanism in promoting solution quality. In terms of computational time, SPEA2-GL is the most computationally expensive algorithm in terms of CPU time whereas MODE-GL is the fastest.

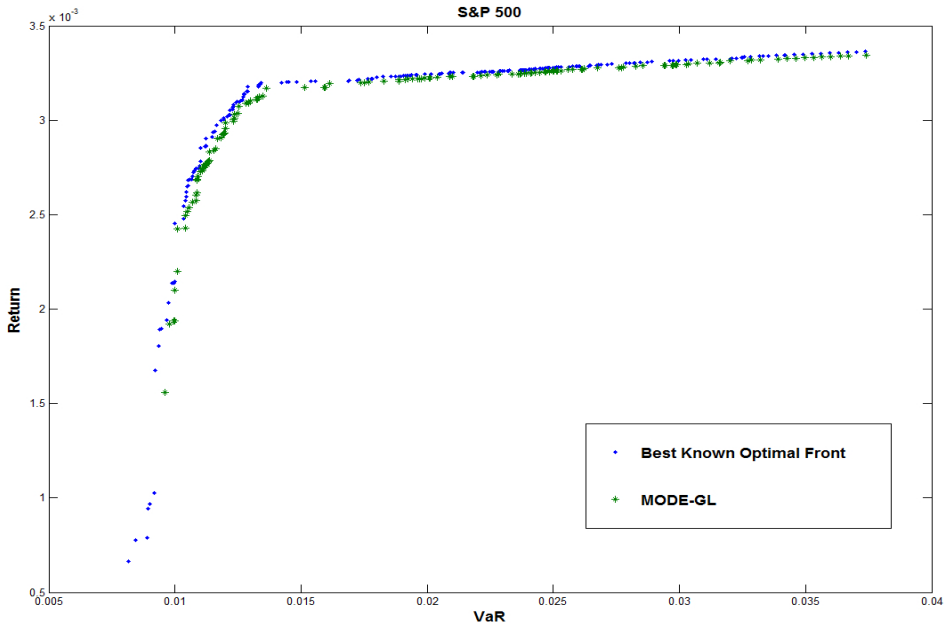
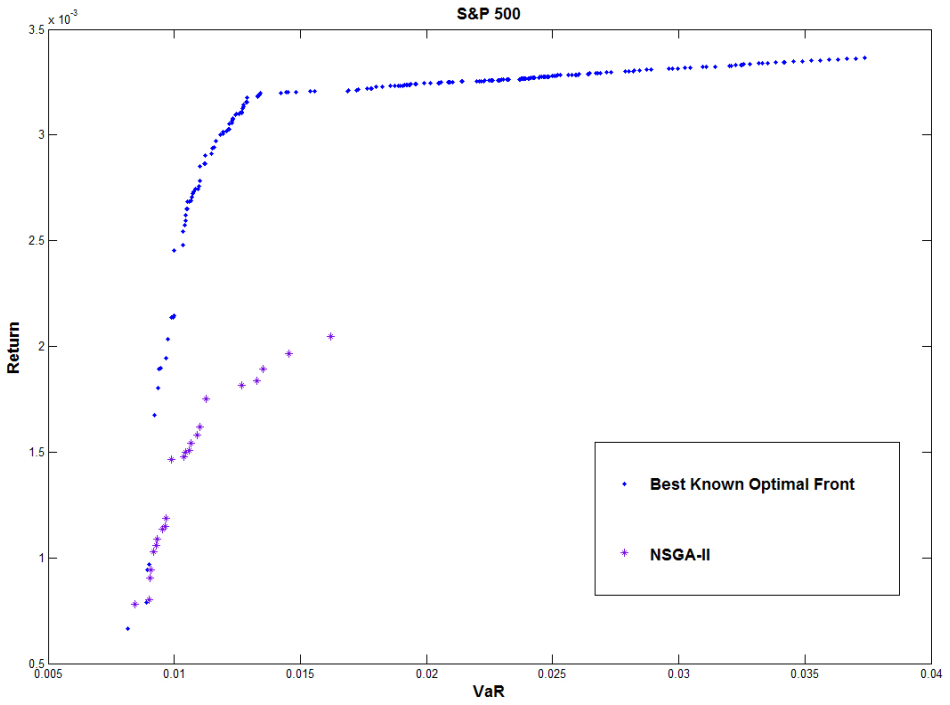


Figure 7.6: S & P 500: Comparison of obtained efficient frontiers of each algorithm together with the best known optimal front from all tested algorithms.



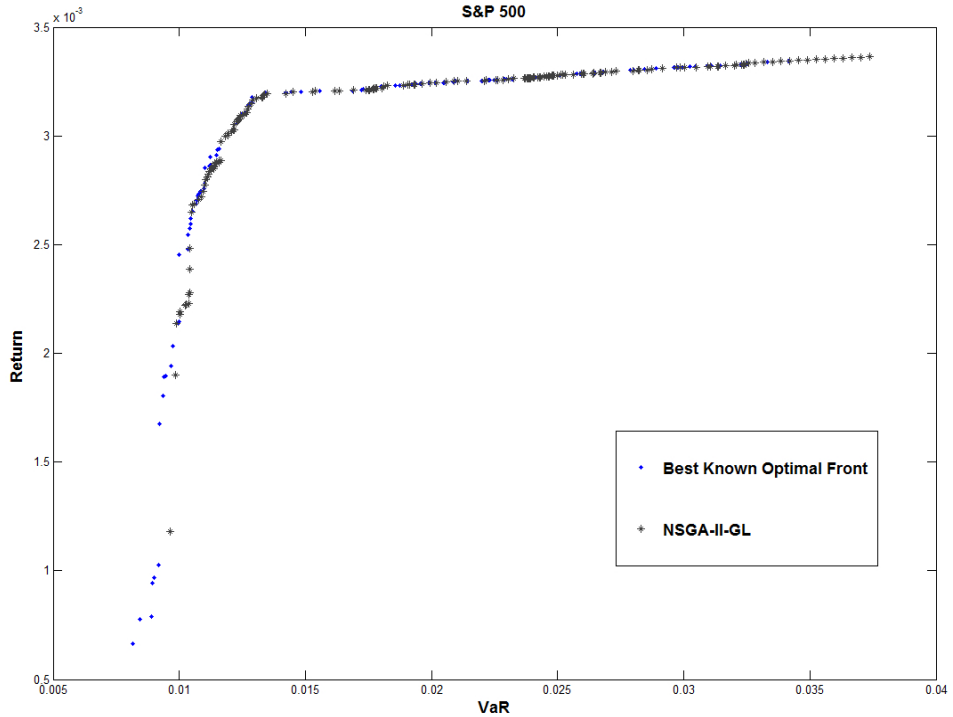


Figure 7.6: S & P 500: Comparison of obtained efficient frontiers of each algorithm together with the best known optimal front from all tested algorithms.

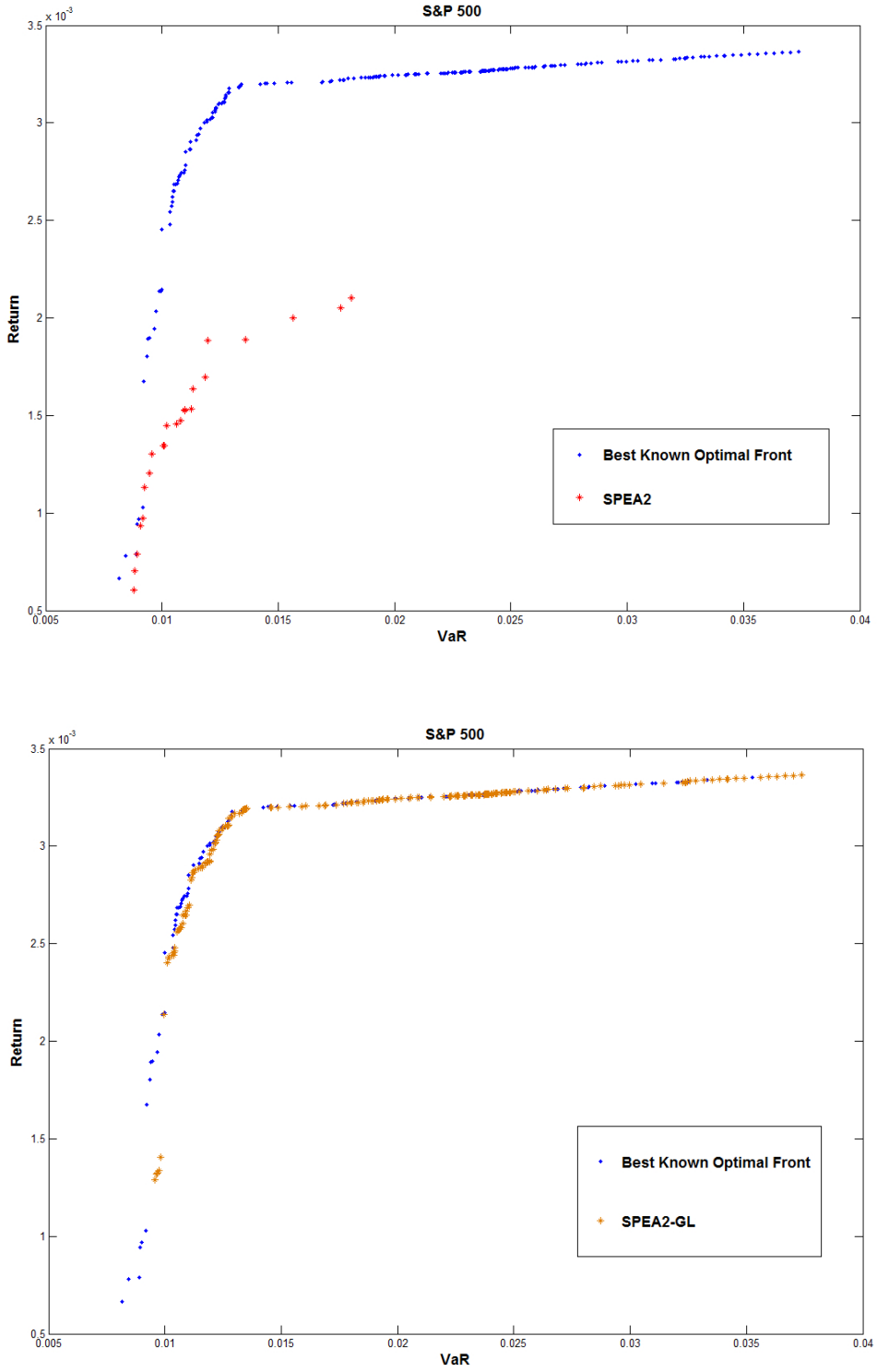


Figure 7.6: S & P 500: Comparison of obtained efficient frontiers of each algorithm together with the best known optimal front from all tested algorithms.

Figure 7.6 provides the obtained efficient frontiers of the tested algorithms for DS2 compared with the best known estimated efficient frontier extracted from all considered algorithms. The horizontal axis describes the loss that might be incurred with a probability $\alpha = 0.05$. Figure 7.6 shows that both NSGA-II and SPEA2 have slow convergence compared to other algorithms.

Algorithm1 $\leftrightarrow$ Algorithm2	IGD	HV
MODE-GL $\leftrightarrow$ NSGA-II	$+$	$+$
MODE-GL $\leftrightarrow$ NSGA-II-GL	$+$	$+$
MODE-GL$\leftrightarrow$ SPEA2	$+$	$+$
MODE-GL $\leftrightarrow$ SPEA2-GL	$+$	$+$
NSGA-II$\leftrightarrow$ NSGA-II-GL	$-$	$-$
NSGA-II $\leftrightarrow$ SPEA2	$\sim$	$\sim$
NSGA-II$\leftrightarrow$ SPEA2-GL	$-$	$-$
NSGA-II-GL $\leftrightarrow$ SPEA2	$+$	$+$
NSGA-II-GL$\leftrightarrow$ SPEA2-GL	$\sim$	$\sim$
SPEA2 $\leftrightarrow$ SPEA2-GL	$-$	$-$

Table 7.2: Student's t-Test Results of Different Algorithms on S & P100 dataset.

We compare the IGD and HV values of the tested algorithms by using Student's t-test (Walpole et al., 1998). The statistical results obtained by a two-tailed t-test with 58 degrees of freedom at a 0.05 level of significance are given in Table 7.2 and Table 7.3.

The results for Algorithm-1$\leftrightarrow$ Algorithm-2 are shown as "$+$", "$-$", or "$\sim$" when Algorithm-1 is significantly better than, significantly worse than, or statistically equivalent to Algorithm-2, respectively. The statistical results reconfirm the effectiveness of the proposed algorithm MODE-GL both in terms of solution quality and computational time. Moreover, the results also show that the performance of the NSGA-II and SPEA2 improves significantly when the learning-guided so-

lution generation scheme is incorporated. Figure 7.7 plots the IGD metric over generation on S & P 100. The results confirm that all the algorithms considered are able to converge.

Algorithm1 $\leftrightarrow$ Algorithm2	IGD	HV
MODE-GL$\leftrightarrow$ NSGA-II	$+$	$+$
MODE-GL $\leftrightarrow$ NSGA-II-GL	$\sim$	$\sim$
MODE-GL $\leftrightarrow$ SPEA2	$+$	$+$
MODE-GL$\leftrightarrow$ SPEA2-GL	$\sim$	$\sim$
NSGA-II $\leftrightarrow$ NSGA-II-GL	$-$	$-$
NSGA-II $\leftrightarrow$ SPEA2	$\sim$	$\sim$
NSGA-II $\leftrightarrow$ SPEA2-GL	$-$	$-$
NSGA-II-GL $\leftrightarrow$ SPEA2	$+$	$+$
NSGA-II-GL $\leftrightarrow$ SPEA2-GL	$\sim$	$\sim$
SPEA2 $\leftrightarrow$ SPEA2-GL	$-$	$-$

Table 7.3: Student's t-Test Results of Different Algorithms on S & P 500 dataset.

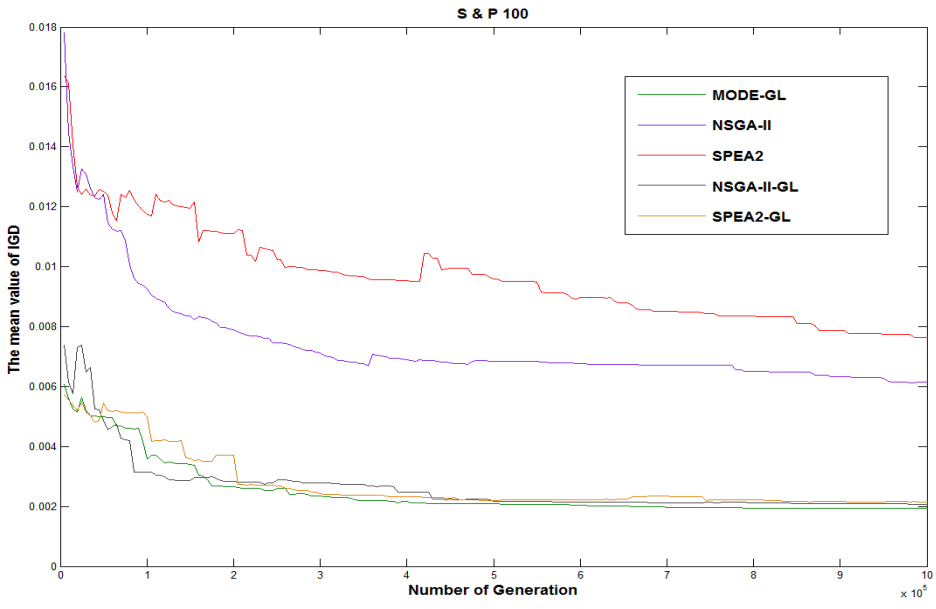


Figure 7.7: Comparison of convergence of algorithms for S & P 100.

7.7 Summary

This chapter presents a multi-objective evolutionary algorithm with guided learning (MODE-GL) for portfolio optimization problems with six widely used practical constraints in real life trading scenarios. This work focuses on downside risk as an alternative risk measure in financial markets and adopts a realistic framework for portfolio optimization that moves away from most widely adopted mean-variance approach. Value-at-Risk (VaR) is used as a risk measure and a historical simulation approach is adopted to calculate VaR. This technique is non-parametric and does not require any distributional assumptions.

The portfolio optimization in the VaR context involves additional complexities since VaR is non-linear, non-convex and non-differentiable, and typically exhibits multiple local extrema and discontinuities especially when real-world trading constraints are incorporated (Gaivoronski and Pflug, 2005). MODE-GL maintains two archives, A and D, and the former is updated with non-dominated solutions and the latter is updated with the least crowded solutions throughout the evolution. A learning mechanism is employed to extract important features from the archive A. These features are exploited in generating promising offspring solutions. MODE-GL thus aims to promote convergence by concentrating on the promising regions of the search space. Two extended variants of differential evolution mutation schemes are utilized in order to promote the exploration of the search towards the least crowded region of the solution space by exploiting the information from the archive D.

The experimental results using real datasets show that MODE-GL outperforms NSGA-II and SPEA2 both in terms of solution quality and computational time. Moreover, the results also show that the performance of the NSGA-II and SPEA2 improves significantly when the learning scheme is incorporated. The MODE-GL approach shows great promise in tackling an important class of portfolio investment problems using realistic constraints in an efficient way and thus has significant potential for adoption in practice.

Chapter 8

Conclusions and Future Work

> "Science is facts; just as houses are made of stone, so is science made of facts; but a pile of stones is not a house, and a collection of facts is not necessarily science."
>
> Jules Henri Poincaré(1854-1912).

This thesis investigates the population-based evolutionary algorithms for single period portfolio optimization problems. The overall objective is to develop efficient and effective evolutionary algorithms and investigate their applications to portfolio optimization problems with additional real-world trading constraints. This thesis can be summarized in two parts. In the first part, the mean-variance portfolio model is investigated by taking into account real-world constraints. In the second part of the thesis, an alternative risk measure, VaR, is considered. A non-parametric mean-VaR model with six practical trading constraints is investigated. Practical trading constraints considered in this thesis are *hard* constraints. Four population-based evolutionary algorithms are presented to efficiently solve these problems (see Table 8.1). We will conclude these in the following sections, which are followed by future work on portfolio optimization problems.

Algorithm	Objective	Constraint	Performance Measure
PBILDE	Single objective optimization by weighted sum method	Cardinality, Quantity	MPE, MedPE
MOSSwA	Multiobjective optimization	Cardinality, Quantity, Pre-assignment	GD, IGD, Spread (Δ)
MODEwAwL	Multiobjective optimization	Cardinality, Quantity, Pre-assignment, Round lot	GD, IGD, Spread (Δ), HV
MODE-GL	Multiobjective optimization	Cardinality, Quantity, Pre-assignment, Round lot, Class, Class Limit	IGD, HV

Table 8.1: Summary of the algorithms with considered constraints.

8.1 Mean Variance Portfolio Optimization

8.1.1 Single Objective Approach

This thesis started with a study of the mean-variance portfolio optimization problem with cardinality and quantity constraints (CCMV). When the cardinality constraint is considered, the model can be viewed as two sub-problems, the determination of the selection of assets and the allocation of capital to each selected asset. Based on PBIL and DE which address the two sub-problems, respectively, a new hybrid evolutionary algorithm (PBILDE) is presented in chapter 4 to efficiently address the CCMV model. A partially guided mutation and an elitist strategy are proposed to enhance the efficient convergence of the search.

The results have shown that PBILDE achieves synergetic effects through hybridization and is competitive with the existing approaches in most problem instances. When cardinality constraints are considered, the lack of efficient selection operations penalizes the performance of the DE. Both PBIL and PBILDE use the probability vector in determining the selection of assets in a portfolio. The results show that DE mutation operator is simpler and more efficient than random allocation following the Gaussian distribution.

PBILDE adopts a single objective optimization approach by aggregating two objectives. As a result, PBILDE suffers from high computational cost since it requires repeated runs of the algorithm in order to obtain the efficient frontier. In addition, it does not consider a good distribution of the obtained solutions nor find Pareto optimal solutions in non-convex regions.

8.1.2 Multi-objective Approach

MOSSwA With Three Trading Constraints

The mean-variance portfolio optimization problem extended with three practical constraints, cardinality, quantity and pre-assignment is studied in chapter 5. A hybrid multi-objective scatter search with an external archive (MOSSwA) algorithm is developed to solve the constrained portfolio problem. Based on the basic template of the scatter search, MOSSwA defines the reference sets based on Pareto dominance and crowding distance measures. It is designed to guide the search toward the Pareto optimal set by exploiting the useful information from the solutions found during the evolution. A new solution combination mechanism is proposed in order to generate the efficient and diverse portfolios. Three problem specific selection heuristics embedded in the solution combination mechanism significantly contribute to the efficient convergence of the MOSSwA.

MOSSwA considers a cardinality constraint which limits the maximum number of assets in a portfolio. From the decision making perspective, a portfolio manager has an extra trade-off criteria (the cardinality of the obtained portfolios) to take into consideration in selecting the suitable portfolio for investment. From a computational perspective, the portfolio problem with the inequality cardinality constraint is less challenging than the equality cardinality constraint. This is due to the fact that feasible space becomes much smaller compared to the search space when strict cardinality constraint is considered.

MODEwAwL With Four Trading Constraints

A new learning-guided multi-objective evolutionary algorithm (MODEwAwL) for the mean-variance portfolio optimization problems is presented in chapter 6. Four real-world constraints, cardinality, quantity, pre-assignment and round lot, are considered. MODEwAwL adopts elitism by maintaining an archive during the evolution process. A learning mechanism is introduced in order to extract important features from the set of elite solutions. An efficient and effective candidate generation scheme utilizing a learning mechanism, problem specific heuristics and effective direction-based search methods is proposed to guide the search towards the promising regions of the search space.

Unlike other MOEAs, MODEwAwL is simple and easy to implement and has very few parameters required to set. Results show that the performance of the evolutionary algorithm can be improved by exploiting problem specific knowledge and learning from the elite solutions encountered during the evolution. When the problem size becomes larger, MODEwAwL converges efficiently while the existing MOEAs suffer from extremely slow convergence due to lack of efficient exploitation of the important features extractable from the solutions throughout the evolution. In addition, extensive experimental results show that MODEwAwL outperforms other existing MOEAs in the literature and can be considered as a promising approach for the problem.

8.2 Mean-VaR Portfolio Optimization

In chapter 7, we study an alternative risk measure, value-at-risk, to replace the variance risk measure in order to better reflect investors' intuition towards the asymmetric return distribution. A historical simulation approach is adopted to calculate VaR. This technique is non-parametric and does not require any distributional assumptions. The portfolio optimization in a mean-VaR framework is a challenging problem since optimizing VaR leads to a non-smooth and non-convex objective function landscape with many local minima. We show that the basic mean-VaR optimization problem becomes a multi-modal problem with many local optima in the feasible space in Figure 7.1. The problem becomes much more challenging when constraints are considered in this framework. In addition, the feasible space becomes much smaller compared to the search space when equality constraints are considered.

A multi-objective evolutionary algorithm with guided learning (MODE-GL) is developed to solve the mean-VaR portfolio optimization problems with six real-world constraints, namely cardinality, quantity, pre-assignment, round lot, class and class limit. The proposed MODE-GL approach extracts the important features of non-dominated solutions throughout the evolution. These features are exploited in generating promising offspring solutions. We find that the basic "DE/rand/1/bin" scheme is inefficient for the considered problem. Two extended variants of differential evolution mutation schemes are therefore proposed in order to promote exploration of the search space to maximize the probability of obtaining the global optimum.

The experimental results using real datasets show that MODE-GL outperforms NSGA-II and SPEA2 both in terms of solution quality and computational time. Moreover, the results also show that the performance of the NSGA-II and SPEA2 improve significantly when the learning scheme is incorporated. The MODE-GL approach shows great promise in tackling an important class of portfolio investment problems using realistic constraints in an efficient way and thus has significant potential for adoption in practice.

8.3 Future Work

For any research project with finite time available to it, there are areas that have not managed to fall into the scope of this thesis but are interesting nevertheless. The following is a non-exhaustive list of future work which would possibly be quite worthy of investigating.

Better Knowledge Exploitation Mechanism

In this thesis, when the cardinality constraints are considered, the learning mechanism is proposed to extract promising assets throughout the evolution. Similarly, a novel learning mechanism may be beneficial by extracting the distributional information of the obtained solution. It would also be interesting to develop a learning scheme which maintains the intervals for the decision variables where good solutions have been found in order to move new solutions towards those intervals (Becerra and Coello, 2006).

Better Constraint handling approaches

In this thesis, all trading constraints are considered as *hard* constraints. Therefore, repair heuristics are adopted in order to generate feasible solutions. It is likely that a repair mechanism may misguide the search process and may result in obtaining poor candidates. In addition, the computational time for repairing the infeasible solutions can be reduced with efficiently designed repair operations (Salcedo-Sanz, 2009). Therefore, it would be beneficial to investigate further for different repair heuristics in order to compare the current repair strategy.

Transaction Cost

In this thesis, we consider up to six practical trading constraints. However, it is still far from reflecting the real market trading scenarios. In financial markets, buying and selling assets entail brokerage fees and taxes imposed on the investors. Transaction cost is one of the main factors concerned by portfolio

managers. Ignoring the transaction cost in the portfolio optimization model may lead to an inefficient portfolio in practice. Therefore, it would be more practical to extend the portfolio optimization model with transaction cost constraint.

Risk Measures

In this thesis, we consider two risk measures: variance and value-at-risk. As noted in Section 7.2, one of the undesirable characteristics of VaR is that it is not a coherent risk measure. Risk quantification for portfolio selection has been actively studied in the literature and many risk measures have been proposed. It would be interesting to consider new coherent risk measures such as expected shortfall and conditional value at risk.

Multi-period Portfolio Optimization

One major criticism of the mean-variance model is the implicit buy-and-hold strategy assumed in the single-period optimization problem. In the ever-changing market conditions, the single-period framework suffers from a short-sighted investment strategy when it is applied repeatedly over many subsequent periods. It is assumed that the return of an asset in each investment period is independently considered. It will be interesting to consider the multi-period portfolio optimization model via dynamic correlation models and/or to enhance the accuracy of the model by replacing historical expected returns with forecasted returns (Chiam et al., 2007; Du et al., 2014).

References

Acerbi, C., Nordio, C., and Sirtori, C. (2001). Expected shortfall as a tool for financial risk management. *arXiv preprint cond-mat/0102304*.

Acerbi, C. and Tasche, D. (2002). On the coherence of expected shortfall. *Journal of Banking & Finance*, 26(7):1487 – 1503.

Adler, I., Resende, M., Veiga, G., and Karmarkar, N. (1989). An Implementation of Karmarkar's Algorithm for Linear Programming. *Mathematical Programming*, 44(1-3):297–335.

Akian, M., Menaldi, J. L., and Sulem, A. (1996). On an Investment-consumption Model with Transaction Costs. *SIAM Journal on Control and Optimization*, 34(1):329–364.

Al Moubayed, N., Petrovski, A., and McCall, J. (2010). A Novel Smart Multi-objective Particle Swarm Optimisation Using Decomposition. In *Parallel Problem Solving from Nature, PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin Heidelberg.

Alexander, G. J. and Baptista, A. M. (2002). Economic Implications of Using a Mean-VaR Model for Portfolio Selection: A Comparison with Mean-Variance Analysis. *Journal of Economic Dynamics and Control*, 26(7-8):1159 – 1193. Finance.

Alfaro-Cid, E., Baixauli-Soler, J. S., and Fernandez-Blanco, M. O. (2011). Minimising Value-at-Risk in a Portfolio Optimisation Problem Using a Multi-

objective Genetic Algorithm. *International Journal of Risk Assessment and Management*, 15(5):453–477.

Anagnostopoulos, K. and Mamanis, G. (2010). A Portfolio Optimization Model with Three Objectives and Discrete Variables . *Computers & Operations Research*, 37(7):1285 – 1297.

Anagnostopoulos, K. and Mamanis, G. (2011a). Multiobjective Evolutionary Algorithms for Complex Portfolio Optimization Problems. *Computational Management Science*, 8(3):259–279.

Anagnostopoulos, K. and Mamanis, G. (2011b). The Mean-Variance Cardinality Constrained Portfolio Optimization Problem: An Experimental Evaluation of Five Multiobjective Evolutionary Algorithms. *Expert Systems with Applications*, 38(11):14208–14217.

Armananzas, R. and Lozano, J. (2005). A Multiobjective Approach to the Portfolio Optimization Problem. In *IEEE Congress on Evolutionary Computation, 2005.*, volume 2, pages 1388–1395 Vol. 2.

Arriaga, J. and Valenzuela-Rendón, M. (2012). Steepest Ascent Hill Climbing for Portfolio Selection. In *Applications of Evolutionary Computation*, volume 7248 of *Lecture Notes in Computer Science*, pages 145–154. Springer Berlin Heidelberg.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999). Coherent Measures of Risk. *Mathematical Finance*, 9(3):203–228.

Bäck, T. and Schwefel, H.-P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary computation*, 1(1):1–23.

Bader, J. and Zitzler, E. (2011). HypE: An Algorithm for Fast Hypervolume-based Many-objective Optimization. *Evolutionary Computation*, 19(1):45–76.

Baixauli-Soler, J., Alfaro-Cid, E., and Fernandez-Blanco, M. (2011). Mean-VaR Portfolio Selection Under Real Constraints. *Computational Economics*, 37(2):113–131.

Baluja, S. (1994). Population-based Incremental Learning. A Method for Integrating Genetic Search based Function Optimization and Competitive Learning. Technical report, DTIC Document.

Baluja, S. (1997). Genetic Algorithms and Explicit Search Statistics. *Advances in Neural Information Processing Systems*, pages 319–325.

Baluja, S. and Caruana, R. (1995). Removing the Genetics from the Standard Genetic Algorithm. In *ICML*, pages 38–46.

Baños, R., Gil, C., Reca, J., and Martínez, J. (2009). Implementation of Scatter Search for Multi-objective Optimization: a Comparative Study. *Computational Optimization and Applications*, 42(3):421–441.

Bawa, V. S. (1975). Optimal Rules for Ordering Uncertain Prospects. *Journal of Financial Economics*, 2(1):95 – 121.

Bawa, V. S. and Lindenberg, E. B. (1977). Capital Market Equilibrium in a Mean-lower Partial Moment Framework . *Journal of Financial Economics*, 5(2):189 – 200.

Beasley, J. (1990). OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society*, 41(11):1069–1072.

Beasley, J. E. (1999). OR Library DataSet. Available from:http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html. [Online; accessed 02-Oct-2011].

Beausoleil, R. P. (2006). "MOSS" Multiobjective Scatter Search Applied to Nonlinear Multiple Criteria Optimization. *European Journal of Operational Research*, 169(2):426 – 449. Feature Cluster on Scatter Search Methods for Optimization.

Becerra, R. L. and Coello, C. A. C. (2006). Cultured Differential Evolution for Constrained Optimization. *Computer Methods in Applied Mechanics and Engineering*, 195(33 - 36):4303 – 4322.

Benati, S. and Rizzi, R. (2007). A Mixed Integer Linear Programming Formulation of the Optimal Mean/Value-at-Risk Portfolio Problem. *European Journal of Operational Research*, 176(1):423 – 434.

Bertsimas, D. and Shioda, R. (2009). Algorithm for Cardinality-constrained Quadratic Optimization. *Computational Optimization and Applications*, 43(1):1–22.

Bienstock, D. (1996). Computational Study of a Family of Mixed-integer Quadratic Programming Problems. *Mathematical Programming*, 74(2):121–140.

Biglova, A., Ortobelli, S., Rachev, S. T., and Stoyanov, S. (2004). Different Approaches to Risk Estimation in Portfolio Theory. *The Journal of Portfolio Management*, 31(1):103–112.

Blum, C., Puchinger, J., Raidl, G. R., and Roli, A. (2011). Hybrid Metaheuristics in Combinatorial Optimization: A Survey . *Applied Soft Computing*, 11(6):4135 – 4151.

Blum, C. and Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308.

Bonissone, P., Subbu, R., Eklund, N., and Kiehl, T. (2006). Evolutionary algorithms + domain knowledge = real-world evolutionary computation. *Evolutionary Computation, IEEE Transactions on*, 10(3):256–280.

Boonma, P. and Suzuki, J. (2011). PIBEA: Prospect Indicator Based Evolutionary Algorithm for Multiobjective Optimization Problems. In *Proc. 2011 IEEE Congress on Evolutionary Computation (CE 2011)*. Citeseer.

Boussaïd, I., Lepagnot, J., and Siarry, P. (2013). A Survey on Optimization Metaheuristics. *Information Sciences*, 237(0):82 – 117. Prediction, Control and Diagnosis using Advanced Neural Computations.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Branke, J., Scheckenbach, B., Stein, M., Deb, K., and Schmeck, H. (2009). Portfolio Optimization with an Envelope-based Multi-objective Evolutionary Algorithm. *European Journal of Operational Research*, 199(3):684 – 693.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., and Zumer, V. (2006). Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657.

Bureerat, S. (2011). Hybrid Population-Based Incremental Learning Using Real Codes. In *Learning and Intelligent Optimization*, volume 6683 of *Lecture Notes in Computer Science*, pages 379–391. Springer Berlin Heidelberg.

Bureerat, S. and Sriworamas, K. (2007). Population-Based Incremental Learning for Multiobjective Optimisation. In *Soft Computing in Industrial Applications*, volume 39 of *Advances in Soft Computing*, pages 223–232. Springer Berlin Heidelberg.

Burke, E. K., Curtois, T., Qu, R., and Berghe, G. V. (2010). A Scatter Search Methodology for the Nurse Rostering Problem. *Journal of the Operational Research Society*, 61(11):1667–1679.

Busetti, F. (2005). Metaheuristic Approaches to Realistic Portfolio Optimization. Master's thesis, University of South Aferica.

Cai, H. R., Chung, C., and Wong, K. (2008). Application of Differential Evolution Algorithm for Transient Stability Constrained Optimal Power Flow. *IEEE Transactions on Power Systems*, 23(2):719–728.

Canakgoz, N. and Beasley, J. (2009). Mixed-integer Programming Approaches for Index Tracking and Enhanced Indexation. *European Journal of Operational Research*, 196(1):384 – 399.

Cesarone, F., Scozzari, A., and Tardella, F. (2011). Portfolio Selection Problems in Practice: A Comparison Between Linear and Quadratic Optimization Models. *arXiv preprint arXiv:1105.3594*.

Cesarone, F., Scozzari, A., and Tardella, F. (2013). A New Method for Mean-Variance Portfolio Optimization with Cardinality Constraints. *Annals of Operations Research*, 205(1):213–234.

Chang, P. C., Chen, S.-H., Zhang, Q., and Lin, J.-L. (2008). MOEA/D for Flowshop Scheduling Problems. In *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence).*, pages 1433–1438.

Chang, T., Meade, N., Beasley, J., and Sharaiha, Y. (2000). Heuristics for Cardinality Constrained Portfolio Optimisation. *Computers and Operations Research*, 27(13):1271–1302.

Chang, T.-J., Yang, S.-C., and Chang, K.-J. (2009). Portfolio Optimization Problems in Different Risk Measures using Genetic Algorithm . *Expert Systems with Applications*, 36(7):10529 – 10537.

Charpentier, A. and Oulidi, A. (2009). Estimating Allocations for Value-at-Risk Portfolio Optimization. *Mathematical Methods of Operations Research*, 69(3):395–410.

Chiam, S., Tan, K., and Al Mamum, A. (2008). Evolutionary Multi-objective Portfolio Optimization in Practical Context. *International Journal of Automation and Computing*, 5(1):67–80.

Chiam, S., Tan, K., and Mamun, A. (2007). Multiobjective Evolutionary Neural Networks for Time Series Forecasting. In *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 346–360. Springer Berlin Heidelberg.

Chopra, V. K. and Ziemba, W. T. (1993). The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice. *The Journal of Portfolio Management*, 19(2):6–11.

Coello, C. A. C. (2002). Theoretical and Numerical Constraint-handling Techniques Used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer methods in Applied Mechanics and Engineering*, 191(11):1245–1287.

Coello, C. A. C., Pulido, G. T., and Lechuga, M. S. (2004). Handling Multiple Objectives with Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279.

Coello, C. A. C. and Zacatenco, C. S. P. (2006). 20 Years of Evolutionary Multiobjective Optimization: What has been done and what remains to be done. *Computational Intelligence: Principles and Practice*, pages 73–88.

Coello, C. C., Lamont, G. B., and van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-objective Problems*. Genetic and Evolutionary Computation. Springer.

Cont, R. (2001). Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues. *Quantitative Finance*, 1(2):223–236.

Corne, D. and Knowles, J. (2003). Some Multiobjective Optimizers are Better Than Others. In *The Congress on Evolutionary Computation, 2003. CEC '03.*, volume 4, pages 2506–2512 Vol.4.

Corne, D., Knowles, J., and Oates, M. (2000). The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization. In *Parallel Problem Solving from Nature PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 839–848. Springer Berlin Heidelberg.

Corne, D. W., Jerram, N. R., Knowles, J. D., Oates, M. J., and J, M. (2001). PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001)*, pages 283–290. Morgan Kaufmann Publishers.

Crama, Y. and Schyns, M. (2003). Simulated Annealing for Complex Portfolio Selection Problems . *European Journal of Operational Research*, 150(3):546 – 571. Financial Modelling.

Cura, T. (2009). Particle Swarm Optimization Approach to Portfolio Optimization . *Nonlinear Analysis: Real World Applications*, 10(4):2396 – 2406.

Dallagnol, V. A. F., van den Berg, J., and Mous, L. (2009). Portfolio Management Using Value at Risk: A Comparison Between Genetic Algorithms and Particle Swarm Optimization. *International Journal of Intelligent Systems*, 24(7):766–792.

Daníelsson, J., Jorgensen, B. N., de Vries, C. G., and Yang, X. (2008). Optimal Portfolio Allocation Under the Probabilistic VaR Constraint and Incentives for Financial Innovation. *Annals of Finance*, 4(3):345–367.

Dantzig, G. (1998). *Linear Programming and Extensions*. Landmarks in Physics and Mathematics. Princeton University Press.

Darwin, C. (1859). *The Origins of Species*. Wordsworth Editions Ltd, published in (1998) edition.

Das, I. and Dennis, J. (1997). A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Structural Optimization*, 14(1):63–69.

Das, S., Abraham, A., and Konar, A. (2008). Automatic Clustering Using an Improved Differential Evolution Algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(1):218–237.

Das, S. and Suganthan, P. (2011). Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31.

Davis, M. H. A. and Norman, A. R. (1990). Portfolio Selection with Transaction Costs. *Mathematics of Operations Research*, 15(4):676–713.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley Interscience Series in Systems and Optimization. Wiley.

Deb, K. (2012). *Optimization for Engineering Design: Algorithms and Examples*. PHI Learning.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Steuer, R., Tewari, R., and Tewari, R. (2011). Bi-objective Portfolio Optimization Using a Customized Hybrid NSGA-II Procedure. In Takahashi, R., Deb, K., Wanner, E., and Greco, S., editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 358–373. Springer Berlin Heidelberg.

Deb, K., Sundar, J., Udaya Bhaskara Rao, N., and Chaudhuri, S. (2006). Reference Point Based Multi-objective Optimization Using Evolutionary Algorithms. *International Journal of Computational Intelligence Research*, 2(3):273–286.

Deng, G.-F., Lin, W.-T., and Lo, C.-C. (2012). Markowitz-based Portfolio Selection with Cardinality Constraints using Improved Particle Swarm Optimization . *Expert Systems with Applications*, 39(4):4558 – 4566.

Di Tollo, G. and Roli, A. (2008). Metaheuristics for the portfolio selection problem. *International Journal of Operations Research*, 5(1):13–35.

Du, W., Leung, S. Y. S., and Kwong, C. K. (2014). Time Series Forecasting by Neural Networks: A Knee Point-based Multiobjective Evolutionary Algorithm Approach . *Expert Systems with Applications*, 41(18):8049 – 8061.

Duffie, D. and Pan, J. (1997). An Overview of Value at Risk. *The Journal of Derivatives*, 4(3):7–49.

Dumas, B. and Luciano, E. (1991). An Exact Solution to a Dynamic Portfolio Choice Problem Under Transactions Costs. *The Journal of Finance*, 46(2):577–595.

Durillo, J., Zhang, Q., Nebro, A., and Alba, E. (2011). Distribution of Computational Effort in ParallelMOEA/D. In Coello, C., editor, *Learning and Intelligent Optimization*, volume 6683 of *Lecture Notes in Computer Science*, pages 488–502. Springer Berlin Heidelberg.

Ehrgott, M., Klamroth, K., and Schwehm, C. (2004). An MCDM Approach to Portfolio Optimization. *European Journal of Operational Research*, 155(3):752 – 770. Traffic and Transportation Systems Analysis.

Fabozzi, F. and Markowitz, H. (2011). *The Theory and Practice of Investment Management: Asset Allocation, Valuation, Portfolio Construction, and Strategies*. Frank J. Fabozzi series. Wiley.

Fama, E. F. (1965). The Behavior of Stock-Market Prices. *The Journal of Business*, 38(1):pp. 34–105.

Fan, H.-Y. and Lampinen, J. (2003). A Trigonometric Mutation Operation to Differential Evolution. *Journal of Global Optimization*, 27(1):105–129.

Fernández, A. and Gómez, S. (2007). Portfolio Selection using Neural Networks . *Computers & Operations Research*, 34(4):1177 – 1191.

Fieldsend, J., Matatko, J., and Peng, M. (2004). Cardinality Constrained Portfolio Optimisation. In Yang, Z., Yin, H., and Everson, R., editors, *Intelligent Data Engineering and Automated Learning–IDEAL 2004*, volume 3177 of *Lecture Notes in Computer Science*, pages 788–793. Springer Berlin Heidelberg.

Fishburn, P. C. (1977). Mean-Risk Analysis with Risk Associated with Below-Target Returns. *The American Economic Review*, 67(2):pp. 116–126.

Folly, K. and Venayagamoorthy, G. (2009). Effects of Learning Rate on the Performance of the Population Based Incremental Learning Algorithm. In *International Joint Conference on Neural Networks, 2009. IJCNN 2009.*, pages 861–868.

Folly, K. A. (2011). Performance Evaluation of Power System Stabilizers based on Population-Based Incremental Learning (PBIL) Algorithm . *International Journal of Electrical Power & Energy Systems*, 33(7):1279 – 1287.

Fonseca, C. M. and Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. In *ICGA*, volume 93, pages 416–423.

Fonseca, C. M. and Fleming, P. J. (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16.

Friedrich, T., Kroeger, T., and Neumann, F. (2011). Weighted Preferences in Evolutionary Multi-objective Optimization. In Wang, D. and Reynolds, M., editors, *AI 2011: Advances in Artificial Intelligence*, volume 7106 of *Lecture Notes in Computer Science*, pages 291–300. Springer Berlin Heidelberg.

Gaivoronski, A. A. and Pflug, G. (2005). Value-at-risk in Portfolio Optimization: Properties and Computational Approach. *Journal of Risk*, 7(2):1–31.

Galić, E. and Höhfeld, M. (1996). Improving the Generalization Performance of Multi-layer-perceptrons with Population-based Incremental Learning. In *Parallel Problem Solving from Nature PPSN IV*, pages 740–750. Springer.

Garcıa López, F., Garcıa Torres, M., Melián Batista, B., Moreno Pérez, J. A., and Moreno-Vega, J. M. (2006). Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research*, 169(2):477–489.

Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Gaspero, L. D., Tollo, G. D., Roli, A., and Schaerf, A. (2011). Hybrid Metaheuristics for Constrained Portfolio Selection Problems. *Quantitative Finance*, 11(10):1473–1487.

Gendreau, M. and Potvin, J.-Y. (2010). *Handbook of Metaheuristics*, volume 2. Springer.

Ghaoui, L. E., Oks, M., and Oustry, F. (2003). Worst-Case Value-At-Risk and Robust Portfolio Optimization: A Conic Programming Approach. *Operations Research*, 51(4):543–556.

Ghosh, A., Das, S., Chowdhury, A., and Giri, R. (2011). An Improved Differential Evolution Algorithm with Fitness-based Adaptation of the Control Parameters . *Information Sciences*, 181(18):3749 – 3765.

Gilli, M. and Këllezi, E. (2002). A Global Optimization Heuristic for Portfolio Choice with VaR and Expected Shortfall. In Kontoghiorghes, E., Rustem, B.,

and Siokos, S., editors, *Computational Methods in Decision-Making, Economics and Finance*, volume 74 of *Applied Optimization*, pages 167–183. Springer US.

Gilli, M., Këllezi, E., and Hysi, H. (2006). A Data-driven Optimization Heuristic for Downside Risk Minimization. *Swiss Finance Institute Research Paper*, (06-2).

Glover, F. (1977). Heuristics for Integer Programming Using Surrogate Constraints. *Decision Sciences*, 8(1):156–166.

Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, 13(5):533–549.

Glover, F. (1998). A Template for Scatter Search and Path Relinking. In *Selected Papers from the Third European Conference on Artificial Evolution*, AE '97, pages 3–54, London, UK, UK. Springer-Verlag.

Glover, F. and Kochenberger, G. (2003). *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Kluwer Academic Publishers.

Glover, F. and Laguna, M. (1998). *Tabu Search*, volume 1. Springer.

Glover, F., Laguna, M., and Martí, R. (2000a). Fundamentals of Scatter Search and Path Relinking. *Control and Cybernetics*, 39(3):653–684.

Glover, F., Løkketangen, A., and Woodruff, D. L. (2000b). Scatter Search to Generate Diverse MIP Solutions. In *Computing Tools for Modeling, Optimization and Simulation*, volume 12 of *Operations Research/Computer Science Interfaces Series*, pages 299–317. Springer US.

Goh, J. W., Lim, K. G., Sim, M., and Zhang, W. (2012). Portfolio Value-at-Risk Optimization for Asymmetrically Distributed Asset Returns . *European Journal of Operational Research*, 221(2):397 – 406.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Golmakani, H. R. and Fazel, M. (2011). Constrained Portfolio Selection using Particle Swarm Optimization . *Expert Systems with Applications*, 38(7):8327 – 8335.

Gosling, T., Jin, N., and Tsang, E. P. (2005). Population Based Incremental Learning with Guided Mutation versus Genetic Algorithms: Iterated Prisoners Dilemma. In *Congress on Evolutionary Computation*, pages 958–965.

Grootveld, H. and Hallerbach, W. (1999). Variance vs Downside Risk: Is There Really that Much Difference? . *European Journal of Operational Research*, 114(2):304 – 319.

Harlow, W. V. (1991). Asset Allocation in a Downside-risk Framework. *Financial Analysts Journal*, pages 28–40.

Herrera, F., Lozano, M., and Molina, D. (2006). Continuous Scatter Search: An Analysis of the Integration of some Combination Methods and Improvement Strategies . *European Journal of Operational Research*, 169(2):450 – 476. Feature Cluster on Scatter Search Methods for Optimization.

Hirschberger, M., Qi, Y., and Steuer, R. E. (2010). Large-scale MV Efficient Frontier Computation via a Procedure of Parametric Quadratic Programming . *European Journal of Operational Research*, 204(3):581 – 588.

Ho, Y. and Pepyne, D. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115(3):549–570.

Hong, Y., Kwong, S., Chang, Y., and Ren, Q. (2008). Unsupervised Feature Selection using Clustering Ensembles and Population Based Incremental Learning Algorithm . *Pattern Recognition*, 41(9):2742 – 2756.

Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2010). Simultaneous Use of Different Scalarizing Functions in MOEA/D. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 519–526, New York, NY, USA. ACM.

Israelsen, C. (2007). The Benefits of Low Correlation. *Journal of Indexes*, 10(6):18–26.

Jackson, P., Maude, D. J., and Perraudin, W. (1997). Bank Capital and Value at Risk. *The Journal of Derivatives*, 4(3):73–89.

Jevne, H., Haddow, P., and Gaivoronski, A. (2012). Evolving Constrained mean-VaR Efficient Frontiers. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8.

Jia, D., Zheng, G., and Khan, M. K. (2011). An Effective Memetic Differential Evolution Algorithm based on Chaotic Local Search . *Information Sciences*, 181(15):3175 – 3187.

Jobst, N., Horniman, M., Lucas, C., and Mitra, G. (2001). Computational Aspects of Alternative Portfolio Selection Models in the Presence of Discrete Asset Choice Constraints. *Quantitative Finance*, 1(5):489–501.

John, C. (2014). *Portfolio Optimization by Heuristic Algorithms*. PhD thesis, School of Computer Science and Electronic Engineering, University of Essex.

Jorion, P. (2006). *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill New York, 3rd edition edition.

Kern, M. (2006). *Parameter Adaptation in Heuristic Search A Population-Based Approach*. PhD thesis, Department of Computer Science, University of Essex.

Kim, I. and de Weck, O. (2005). Adaptive Weighted-sum Method for Bi-objective Optimization: Pareto Front Generation. *Structural and Multidisciplinary Optimization*, 29(2):149–158.

Kim, Y., Giacometti, R., Rachev, S., Fabozzi, F., and Mignacca, D. (2012). Measuring Financial Risk and Portfolio Optimization with a Non-Gaussian Multivariate Model. *Annals of Operations Research*, 201(1):325–343.

Knowles, J. and Corne, D. (2002). On Metrics for Comparing Nondominated Sets. In *Proceedings of the Congress on Evolutionary Computation, 2002. CEC '02.*, volume 1, pages 711–716.

Knowles, J., Thiele, L., and Zitzler, E. (2006). A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland. Revised Version.

Knowles, J. D. and Corne, D. W. (2000). Approximating the Nondominated Front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172.

Kolm, P. N., Tütüncü, R., and Fabozzi, F. J. (2014). 60 Years of Portfolio Optimization: Practical Challenges and Current Trends . *European Journal of Operational Research*, 234(2):356 – 371.

Konak, A., Coit, D. W., and Smith, A. E. (2006). Multi-objective Optimization using Genetic Algorithms: A Tutorial . *Reliability Engineering & System Safety*, 91(9):992 – 1007. Special Issue - Genetic Algorithms and Reliability Special Issue - Genetic Algorithms and Reliability.

Konstantinidis, A. and Yang, K. (2011). Multi-objective Energy-efficient Dense Deployment in Wireless Sensor Networks using a Hybrid Problem-specific MOEA/D . *Applied Soft Computing*, 11(6):4117 – 4134.

Krink, T., Mittnik, S., and Paterlini, S. (2009). Differential Evolution and Combinatorial Search for Constrained Index-tracking. *Annals of Operations Research*, 172(1):153–176.

Krink, T. and Paterlini, S. (2011). Multiobjective Optimization using Differential Evolution for Real-world Portfolio Optimization. *Computational Management Science*, 8(1-2):157–179.

Krokhmal, P., Zabarankin, M., and Uryasev, S. (2011). Modeling and Optimization of Risk . *Surveys in Operations Research and Management Science*, 16(2):49 – 66.

Lampinen, J. (2002). A Constraint Handling Approach for the Differential Evolution Algorithm. In *Computational Intelligence, Proceedings of the World on Congress on*, volume 2, pages 1468–1473. IEEE.

Larrañaga, P. and Lozano, J. (2002). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Springer US.

Li, D., Sun, X., and Wang, J. (2006). Optimal Lot Solution to Cardinality Constrained Mean–Variance Formulation for Portfolio Selection. *Mathematical Finance*, 16(1):83–101.

Li, H. and Landa-Silva, D. (2008). Evolutionary Multi-objective Simulated Annealing with Adaptive and Competitive Search Direction. In *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence).*, pages 3311–3318.

Li, H. and Landa-Silva, D. (2011). An Adaptive Evolutionary Multi-Objective Approach Based on Simulated Annealing. *Evolutionary Computation*, 19(4):561–595.

Li, H. and Zhang, Q. (2009). Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302.

Liagkouras, K. and Metaxiotis, K. (2014). A New Probe Guided Mutation Operator and its Application for Solving the Cardinality Constrained Portfolio Optimization Problem . *Expert Systems with Applications*, 41(14):6274 – 6290.

Lin, C.-C. and Liu, Y.-T. (2008). Genetic Algorithms for Portfolio Selection Problems with Minimum Transaction Lots . *European Journal of Operational Research*, 185(1):393 – 404.

Linsmeier, T. J. and Pearson, N. D. (2000). Value at Risk. *Financial Analysts Journal*, pages 47–67.

Liu, H., Cai, Z., and Wang, Y. (2010). Hybridizing Particle Swarm Optimization with Differential Evolution for Constrained Numerical and Engineering Optimization . *Applied Soft Computing*, 10(2):629 – 640.

Lobo, M., Fazel, M., and Boyd, S. (2007). Portfolio Optimization with Linear and Fixed Transaction Costs. *Annals of Operations Research*, 152(1):341–365.

Lwin, K. and Qu, R. (2013). A Hybrid Algorithm for Constrained Portfolio Selection Problems. *Applied Intelligence*, 39(2):251–266.

Lwin, K., Qu, R., and Jianhua, Z. (2013). Multi-objective Scatter Search with External Archive for Portfolio Optimization. 5th International Conference on Evolutionary Computation Theory and Applications, pages 111–119, Vilamoura, Algrave, Portugal.

Lwin, K., Qu, R., and Kendall, G. (2014). A Learning-guided Multi-objective Evolutionary Algorithm for Constrained Portfolio Optimization . *Applied Soft Computing*, 24(0):757 – 772.

Maenhout, B. and Vanhoucke, M. (2010). A Hybrid Scatter Search Heuristic for Personalized Crew Rostering in the Airline Industry . *European Journal of Operational Research*, 206(1):155 – 167.

Manganelli, S. and Engle, R. F. (2001). Value at Risk Models in Finance. Technical report, European Central Bank Frankfurt am Main.

Mansini, R., Ogryczak, W., and Speranza, M. G. (2014). Twenty years of linear programming based portfolio optimization. *European Journal of Operational Research*, 234(2):518 – 535. 60 years following Harry Markowitzs contribution to portfolio theory and operations research.

Mansini, R. and Speranza, M. G. (1999). Heuristic Algorithms for the Portfolio Selection Problem with Minimum Transaction Lots . *European Journal of Operational Research*, 114(2):219 – 233.

Maringer, D. (2005). *Portfolio Management with Heuristic Optimization*, volume 8 of *Advances in Computational Management Science*. Springer Verlag.

Maringer, D. and Kellerer, H. (2003). Optimization of Cardinality Constrained Portfolios with a Hybrid Local Search Algorithm. *OR Spectrum*, 25(4):481–495.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1):pp. 77–91.

Markowitz, H. (1956). The Optimization of a Quadratic Function Subject to Linear Constraints. *Naval Research Logistics Quarterly*, 3(1-2):111–133.

Markowitz, H. (1959). Portfolio Selection: Efficient Diversification of Investments. *John Wiley and Sons, New York*.

Markowitz, H. M. (1987). Mean-variance Analysis in Portfolio Choice and Capital Markets.

Marler, R. and Arora, J. (2010). The Weighted Sum Method for Multi-objective Optimization: New Insights. *Structural and Multidisciplinary Optimization*, 41(6):853–862.

Martí, R., Laguna, M., and Glover, F. (2006). Principles of Scatter Search . *European Journal of Operational Research*, 169(2):359 – 372. Feature Cluster on Scatter Search Methods for Optimization.

McDonnell, P. (2008). *Optimal Portfolio Modeling: Models to Maximize Returns and Control Risk in Excel and R*. Wiley Trading. Wiley.

Merton, R. C. (1972). An Analytic Derivation of the Efficient Portfolio Frontier. *Journal of Financial and Quantitative Analysis*, 7(04):1851–1872.

Metaxiotis, K. and Liagkouras, K. (2012). Multiobjective Evolutionary Algorithms for Portfolio Management: A Comprehensive Literature Review . *Expert Systems with Applications*, 39(14):11685 – 11698.

Mezura-Montes, E. (2009). *Constraint-Handling in Evolutionary Optimization*. Studies in Computational Intelligence. Springer.

Mezura-Montes, E., Miranda-Varela, M. E., and del Carmen Gómez-Ramón, R. (2010). Differential Evolution in Constrained Numerical Optimization: An Empirical Study. *Information Sciences*, 180(22):4223 – 4262.

Mezura-Montes, E. n., Velázquez-Reyes, J., and Coello Coello, C. A. (2006). A Comparative Study of Differential Evolution Variants for Global Optimization. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 485–492, New York, NY, USA. ACM.

Michaud, R. O. and Michaud, R. O. (2008). *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. Oxford University Press.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Springer US.

Mininno, E., Neri, F., Cupertino, F., and Naso, D. (2011). Compact Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 15(1):32–54.

Mishra, S. K., Panda, G., and Majhi, R. (2014). A Comparative Performance Assessment of a Set of Multiobjective Algorithms for Constrained Portfolio Assets Selection. *Swarm and Evolutionary Computation*, 16(0):38 – 51.

Mohamed, A. W. and Sabry, H. Z. (2012). Constrained Optimization based on Modified Differential Evolution Algorithm . *Information Sciences*, 194(0):171 – 208. Intelligent Knowledge-Based Models and Methodologies for Complex Information Systems.

Moral-Escudero, R., Ruiz-Torrubiano, R., and Suarez, A. (2006). Selection of Optimal Investment Portfolios with Cardinality Constraints. In *IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, pages 2382–2388.

Morgan, J. P. (1996). RiskMetrics (TM): Technical Document, 4th ed. *New York: Morgan Guaranty Trust Company*.

Moscato, P., Cotta, C., and Mendes, A. (2004). Memetic Algorithms. In *New Optimization Techniques in Engineering*, volume 141 of *Studies in Fuzziness and Soft Computing*, pages 53–85. Springer Berlin Heidelberg.

Natarajan, K., Pachamanova, D., and Sim, M. (2008). Incorporating Asymmetric Distributional Information in Robust Value-at-Risk Optimization. *Management Science*, 54(3):573–585.

Nebro, A., Luna, F., Alba, E., Dorronsoro, B., Durillo, J., and Beham, A. (2008). AbYSS: Adapting Scatter Search to Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 12(4):439–457.

Neri, F. and Tirronen, V. (2010). Recent Advances in Differential Evolution: A Survey and Experimental Analysis. *Artificial Intelligence Review*, 33(1-2):61–106.

Niedermayer, A. and Niedermayer, D. (2010). Applying Markowitzs Critical Line Algorithm. In *Handbook of Portfolio Construction*, pages 383–400. Springer.

Noman, N. and Iba, H. (2008). Accelerating Differential Evolution Using an Adaptive Local Search. *IEEE Transactions on Evolutionary Computation*, 12(1):107–125.

Oksendal, B. and Sulem, A. (2002). Optimal Consumption and Portfolio with Both Fixed and Proportional Transaction Costs. *SIAM Journal on Control and Optimization*, 40(6):1765–1790.

Onwubolu, G. and Davendra, D. (2006). Scheduling Flow Shops using Differential Evolution Algorithm. *European Journal of Operational Research*, 171(2):674 – 692.

Peng, W., Zhang, Q., and Li, H. (2009). Comparison between MOEA/D and NSGA-II on the Multi-Objective Travelling Salesman Problem. In Goh, C.-K., Ong, Y.-S., and Tan, K., editors, *Multi-Objective Memetic Algorithms*, volume 171 of *Studies in Computational Intelligence*, pages 309–324. Springer Berlin Heidelberg.

Pérignon, C. and Smith, D. R. (2010). The Level and Quality of Value-at-Risk Disclosure by Commercial Banks . *Journal of Banking & Finance*, 34(2):362 – 377.

Pholdee, N. and Bureerat, S. (2013). Hybridisation of Real-code Population-based Incremental Learning and Differential Evolution for Multiobjective Design of Trusses . *Information Sciences*, 223(0):136 – 152.

Pinol, H. and Beasley, J. (2006). Scatter Search and Bionomic Algorithms for the Aircraft Landing Problem . *European Journal of Operational Research*, 171(2):439 – 462.

Ponsich, A., Jaimes, A., and Coello, C. (2013). A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications. *IEEE Transactions on Evolutionary Computation*, 17(3):321–344.

Prakash, A. J., Chang, C.-H., and Pactwa, T. E. (2003). Selecting a portfolio with skewness: Recent evidence from US, European, and Latin American equity markets . *Journal of Banking & Finance*, 27(7):1375 – 1390.

Price, K., Storn, R., and Lampinen, J. (2006). *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer London, Limited.

Qin, A. K., Huang, V. L., and Suganthan, P. (2009). Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 13(2):398–417.

Quek, H.-Y., Chen Tan, K., Goh, C.-K., and Abbass, H. (2009). Evolution and Incremental Learning in the Iterated Prisoner's Dilemma. *IEEE Transactions on Evolutionary Computation*, 13(2):303–320.

Rachmawati, L. and Srinivasan, D. (2006). Preference Incorporation in Multiobjective Evolutionary Algorithms: A Survey. In *IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, pages 962–968.

Rahnamayan, S., Tizhoosh, H., and Salama, M. (2008). Opposition-Based Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 12(1):64–79.

Raidl, G. R. (2006). A Unified View on Hybrid Metaheuristics. In *Hybrid Metaheuristics*, volume 4030 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg.

Resende, M., Ribeiro, C., Glover, F., and Martí, R. (2010). Scatter Search and Path-Relinking: Fundamentals, Advances, and Applications. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 87–107. Springer US.

Robič, T. and Filipič, B. (2005). DEMO: Differential Evolution for Multiobjective Optimization. In Coello Coello, C., Hernndez Aguirre, A., and Zitzler, E., editors, *Evolutionary Multi-Criterion Optimization*, volume 3410 of *Lecture Notes in Computer Science*, pages 520–533. Springer Berlin Heidelberg.

Rockafellar, R. and Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443 – 1471.

Rockafellar, R. T. and Uryasev, S. (2000). Optimization of Conditional Value-at-Risk. *Journal of Risk*, 2:21–42.

Roy, A. D. (1952). Safety First and the Holding of Assets. *Econometrica: Journal of the Econometric Society*, pages 431–449.

Ruiz-Torrubiano, R. and Suarez, A. (2010). Hybrid Approaches and Dimensionality Reduction for Portfolio Selection with Cardinality Constraints. *IEEE Computational Intelligence Magazine*, 5(2):92–107.

Russell, R. A. and Chiang, W.-C. (2006). Scatter Search for the Vehicle Routing Problem with Time Windows . *European Journal of Operational Research*, 169(2):606 – 622.

Salcedo-Sanz, S. (2009). A Survey of Repair Methods used as Constraint Handling Techniques in Evolutionary Algorithms. *Computer Science Review*, 3(3):175 – 192.

Schaerf, A. (2002). Local Search Techniques for Constrained Portfolio Selection Problems. *Computational Economics*, 20(3):177–190.

Scherer, B. and Martin, R. (2005). *Modern Portfolio Optimization with NuOPT, S-PLUS®, and S+Bayes*. Springer.

Sebag, M. and Ducoulombier, A. (1998). Extending Population-based Incremental Learning to Continuous Search Spaces. In Eiben, A., Bck, T., Schoenauer, M., and Schwefel, H.-P., editors, *Parallel Problem Solving from NaturePPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 418–427. Springer Berlin Heidelberg.

Shapiro, J. (2003). The Sensitivity of PBIL to its Learning Rate and How Detailed Balance Can Remove It . *Foundations of Genetic Algorithms*, 7:115–132.

Shaw, D. X., Liu, S., and Kopman, L. (2008). Lagrangian Relaxation Procedure for Cardinality-constrained Portfolio Optimization. *Optimisation Methods & Software*, 23(3):411–420.

Shreve, S. E. and Soner, H. M. (1994). Optimal Investment and Consumption with Transaction Costs. *The Annals of Applied Probability*, 4(3):pp. 609–692.

Sierra, M. R. and Coello Coello, C. A. (2005). Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and Epsilon-Dominance. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, EMO'05, pages 505–519, Berlin, Heidelberg. Springer-Verlag.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Simon, D. (2013). *Evolutionary Optimization Algorithms*. Wiley.

Skolpadungket, P., Dahal, K., and Harnpornchai, N. (2007). Portfolio Optimization using Multi-objective Genetic Algorithms. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 516–523.

Smith, E. D., Son, Y. J., Piattelli-Palmarini, M., and Terry Bahill, A. (2007). Ameliorating Mental Mistakes in Tradeoff Studies. *Systems Engineering*, 10(3):222–240.

Soleimani, H., Golmakani, H. R., and Salimi, M. H. (2009). Markowitz-based Portfolio Selection with Minimum Transaction Lots, Cardinality Constraints and Regarding Sector Capitalization using Genetic Algorithm . *Expert Systems with Applications*, 36(3, Part 1):5058 – 5063.

Southey, F. and Karray, F. (1999). Approaching Evolutionary Robotics Through Population-based Incremental Learning. In *Systems, Man, and Cybernetics,*

1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on, volume 2, pages 710–715 vol.2.

Srinivas, N. and Deb, K. (1994). Muiltiobjective Optimization using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248.

Stein, M., Branke, J., and Schmeck, H. (2008). Efficient Implementation of an Active Set Algorithm for Large-scale Portfolio Selection . *Computers & Operations Research*, 35(12):3945 – 3961. Part Special Issue: Telecommunications Network Engineering.

Storn, R. and Price, K. (1997). Differential Evolution–a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359.

Stoyanov, S., Rachev, S., and Fabozzi, F. (2013). Sensitivity of Portfolio VaR and CVaR to Portfolio Return Characteristics. *Annals of Operations Research*, 205(1):169–187.

Streichert, F., Ulmer, H., and Zell, A. (2004a). Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem. In Deb, K., editor, *Genetic and Evolutionary Computation – GECCO 2004*, volume 3103 of *Lecture Notes in Computer Science*, pages 1239–1250. Springer Berlin Heidelberg.

Streichert, F., Ulmer, H., and Zell, A. (2004b). Evaluating a Hybrid Encoding and Three Crossover Operators on the Constrained Portfolio Selection Problem. In *Congress on Evolutionary Computation, 2004. CEC2004.*, volume 1, pages 932–939 Vol.1.

Streichert, F., Ulmer, H., and Zell, A. (2004c). Evolutionary Algorithms and the Cardinality Constrained Portfolio Optimization Problem. volume 2003 of *Operations Research Proceedings*, pages 253–260. Springer Berlin Heidelberg.

Sun, J., Zhang, Q., and Tsang, E. P. (2005). DE/EDA: A New Evolutionary Algorithm for Global Optimization . *Information Sciences*, 169(34):249 – 262.

Talbi, E.-G. (2002). A Taxonomy of Hybrid Metaheuristics. *Journal of Heuristics*, 8(5):541–564.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley.

Tan, K., Khor, E., and Lee, T. (2006). *Multiobjective Evolutionary Algorithms and Applications: Algorithms and Applications*. Advanced Information and Knowledge Processing. Springer.

Tapia, M. G. C. and Coello, C. A. C. (2007). Applications of Multi-objective Evolutionary Algorithms in Economics and Finance: A Survey. In *IEEE Congress on Evolutionary Computation*, volume 7, pages 532–539.

Tasoulis, D., Pavlidis, N., Plagianakos, V., and Vrahatis, M. (2004). Parallel Differential Evolution. In *Congress on Evolutionary Computation, 2004. CEC2004.*, volume 2, pages 2023–2029 Vol.2.

Thiele, L., Miettinen, K., Korhonen, P. J., and Molina, J. (2009). A Preference-based Evolutionary Algorithm for Multi-objective Optimization. *Evolutionary Computation*, 17(3):411–436.

Van Veldhuizen, D. and Lamont, G. (2000). On Measuring Multiobjective Evolutionary Algorithm Performance. In *Proceedings of the Congress on Evolutionary Computation, 2000.*, pages 204–211 vol.1.

Van Veldhuizen, D. A. and Lamont, G. B. (1998). Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical report.

Vasile, M., Minisci, E., and Locatelli, M. (2011). An Inflationary Differential Evolution Algorithm for Space Trajectory Optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):267–281.

Vega-Rodríguez, M. A., Vega-Pérez, D., Gómez-Pulido, J. A., and Sánchez-Pérez, J. M. (2007). Radio Network Design Using Population-Based Incremental Learning and Grid Computing with BOINC. In *Applications of Evolutionary Computing*, volume 4448 of *Lecture Notes in Computer Science*, pages 91–100. Springer Berlin Heidelberg.

Ventresca, M. and Tizhoosh, H. R. (2008). A Diversity Maintaining Population-based Incremental Learning Algorithm . *Information Sciences*, 178(21):4038 – 4056.

Vielma, J. P., Ahmed, S., and Nemhauser, G. L. (2008). A Lifted Linear Programming Branch-and-bound Algorithm for Mixed-integer Conic Quadratic Programs. *INFORMS Journal on Computing*, 20(3):438–450.

Vijayalakshmi Pai, G. and Michel, T. (2009). Evolutionary Optimization of Constrained k-Means Clustered Assets for Diversification in Small Portfolios. *IEEE Transactions on Evolutionary Computation*, 13(5):1030–1053.

Vince, R. (2007). *The Handbook of Portfolio Mathematics: Formulas for Optimal Allocation & Leverage*, volume 257. John Wiley & Sons Inc.

Walpole, R. E., Myers, R. H., Myers, S. L., and Ye, K. (1998). *Probability and Statistics for Engineers and Scientists*, volume 8. Prentice Hall Upper Saddle River, NJ:.

Wang, Y. and Cai, Z. (2012). Combining Multiobjective Optimization With Differential Evolution to Solve Constrained Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 16(1):117–134.

Wang, Y., Li, B., and Weise, T. (2010). Estimation of Distribution and Differential Evolution Cooperation for Large Scale Economic Load Dispatch Optimization of Power Systems . *Information Sciences*, 180(12):2405 – 2420.

Weigand, R. A. (2014). *Applied Equity Analysis and Portfolio Management + Online Video Course: Tools to Analyze and Manage Your Stock Portfolio*. Wiley Finance. John Wiley & Sons.

Winker, P., Lyra, M., and Sharpe, C. (2011). Least Median of Squares Estimation by Optimization Heuristics with an Application to the CAPM and a Multi-factor Model. *Computational Management Science*, 8(1-2):103–123.

Wolfe, P. (1959). The Simplex Method for Quadratic Programming. *Econometrica*, 27(3):pp. 382–398.

Wolpert, D. and Macready, W. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Woodside-Oriakhi, M., Lucas, C., and Beasley, J. (2011). Heuristic Algorithms for the Cardinality Constrained Efficient Frontier . *European Journal of Operational Research*, 213(3):538 – 550.

Xing, H. and Qu, R. (2011a). A Population Based Incremental Learning for Delay Constrained Network Coding Resource Minimization. In *Applications of Evolutionary Computation*, pages 51–60. Springer.

Xing, H. and Qu, R. (2011b). A Population Based Incremental Learning for Network Coding Resources Minimization. *IEEE Communications Letters*, 15(7):698–700.

Xu, R.-t., Zhang, J., Liu, O., and Huang, R.-z. (2010). An Estimation of Distribution Algorithm Based Portfolio Selection Approach. In *Proceedings of International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 305–313.

Yang, S., Ho, S., Ni, G., Machado, J., and Wong, K. F. (2007). A New Implementation of Population Based Incremental Learning Method for Optimizations in Electromagnetics. *IEEE Transactions on Magnetics*, 43(4):1601–1604.

Yang, S. and Yao, X. (2005). Experimental Study on Population-based Incremental Learning Algorithms for Dynamic Optimization Problems. *Soft Computing*, 9(11):815–834.

Yang, Z., Tang, K., and Yao, X. (2008). Large Scale Evolutionary Optimization using Cooperative Coevolution. *Information Sciences*, 178(15):2985 – 2999.

Yuan, B. and Gallagher, M. (2003). Playing in Continuous Spaces: Some Analysis and Extension of Population-based Incremental Learning. In *Proceedings of Congress on Evolutionary Computation, 2003. CEC '03.*, volume 1, pages 443–450 Vol.1.

Zhang, G., Cheng, J., Gheorghe, M., and Meng, Q. (2013). A Hybrid Approach based on Differential Evolution and Tissue Membrane Systems for Solving Constrained Manufacturing Parameter Optimization Problems . *Applied Soft Computing*, 13(3):1528 – 1542.

Zhang, J. and Sanderson, A. (2009). JADE: Adaptive Differential Evolution With Optional External Archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958.

Zhang, M., Luo, W., and Wang, X. (2008). Differential Evolution with Dynamic Stochastic Selection for Constrained Optimization . *Information Sciences*, 178(15):3043 – 3074.

Zhang, Q. and Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

Zhang, Q., Li, H., Maringer, D., and Tsang, E. (2010). MOEA/D with NBI-style Tchebycheff Approach for Portfolio Management. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.

Zhang, Q., Sun, J., and Tsang, E. (2005). An Evolutionary Algorithm with Guided Mutation for the Maximum Clique Problem. *IEEE Transactions on Evolutionary Computation*, 9(2):192–200.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective Evolutionary Algorithms: A Survey of the State of the Art . *Swarm and Evolutionary Computation*, 1(1):32 – 49.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary computation*, 8(2):173–195.

Zitzler, E. and Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer Berlin Heidelberg.

Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L., and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm.

Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Zitzler, E., Thiele, L., and Bader, J. (2010). On Set-Based Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 14(1):58–79.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., and da Fonseca, V. (2003). Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.

Appendix A

A.1 Additional Experimental Test1

This section provides the additional computational results of five MOEAs considered in chapter 6 by the following parameter values.

Constraint values: $\mathcal{K} = 5$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z = \{30\}$, $\vartheta_i = 0.008$.

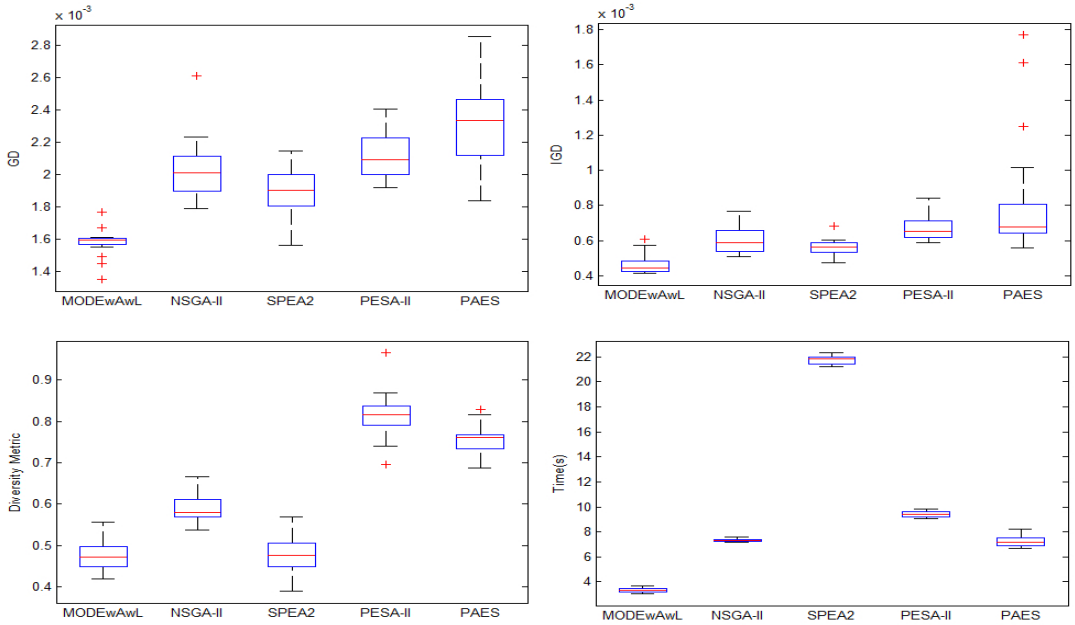


Figure A.1: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Hang Seng dataset with $\mathcal{K} = 5$.

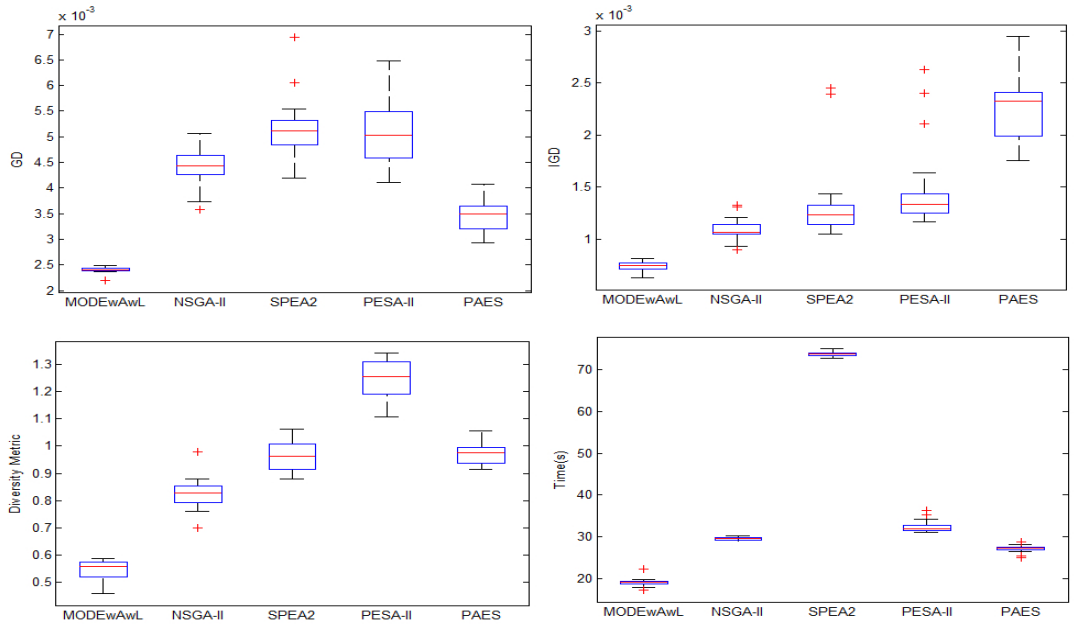


Figure A.2: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for DAX 100 dataset $\mathcal{K} = 5$.

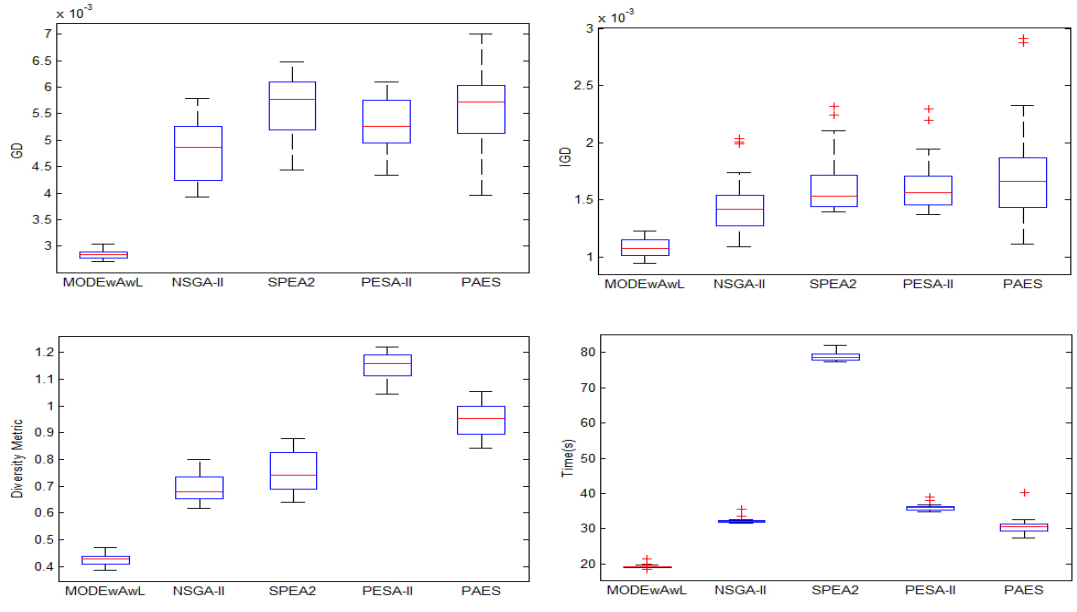


Figure A.3: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for FTSE 100 dataset $\mathcal{K} = 5$.

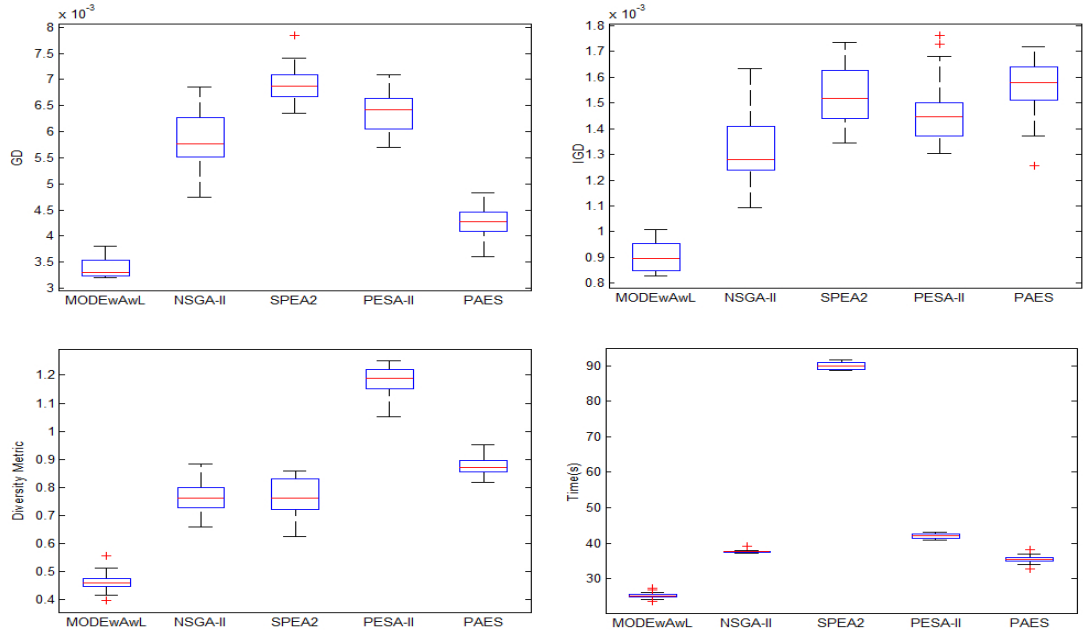


Figure A.4: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for S & P 100 dataset $\mathcal{K} = 5$.

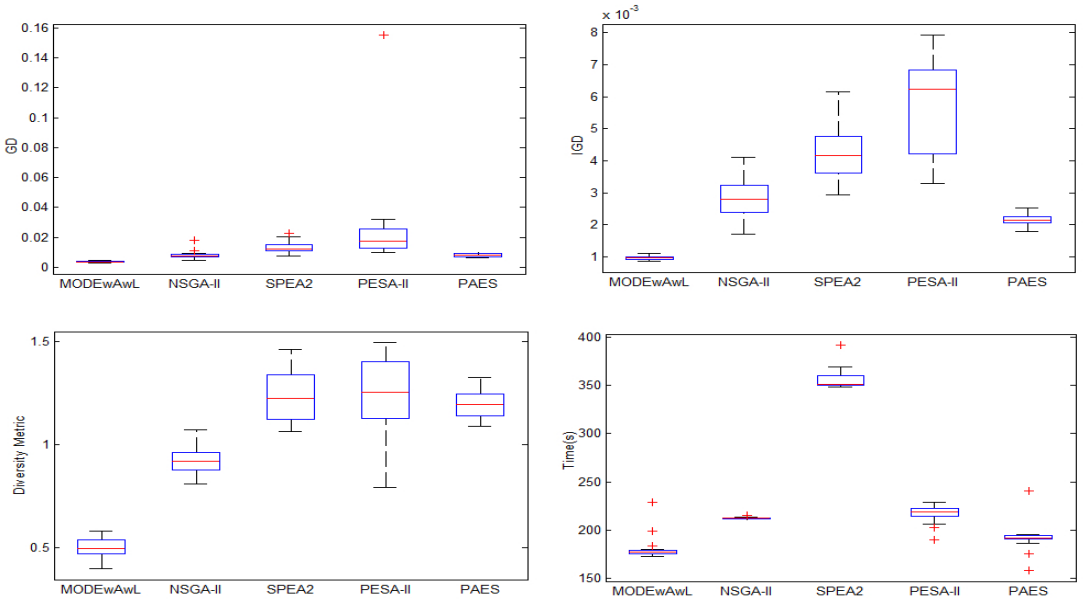


Figure A.5: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) Metrics for Nikkei dataset $\mathcal{K} = 5$.

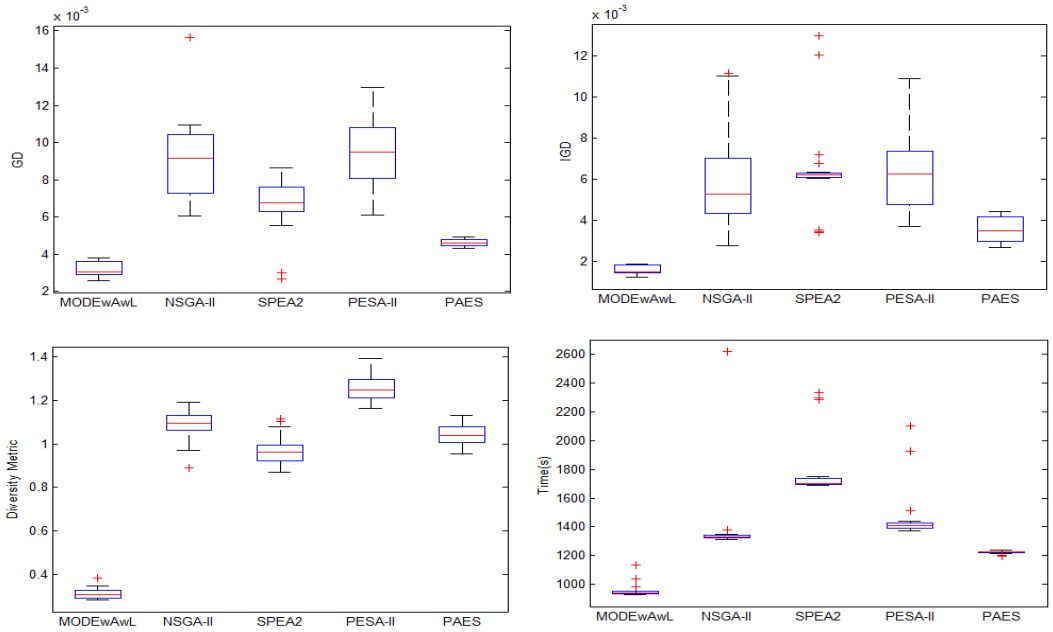


Figure A.6: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for S & P 500 dataset $\mathcal{K} = 5$.

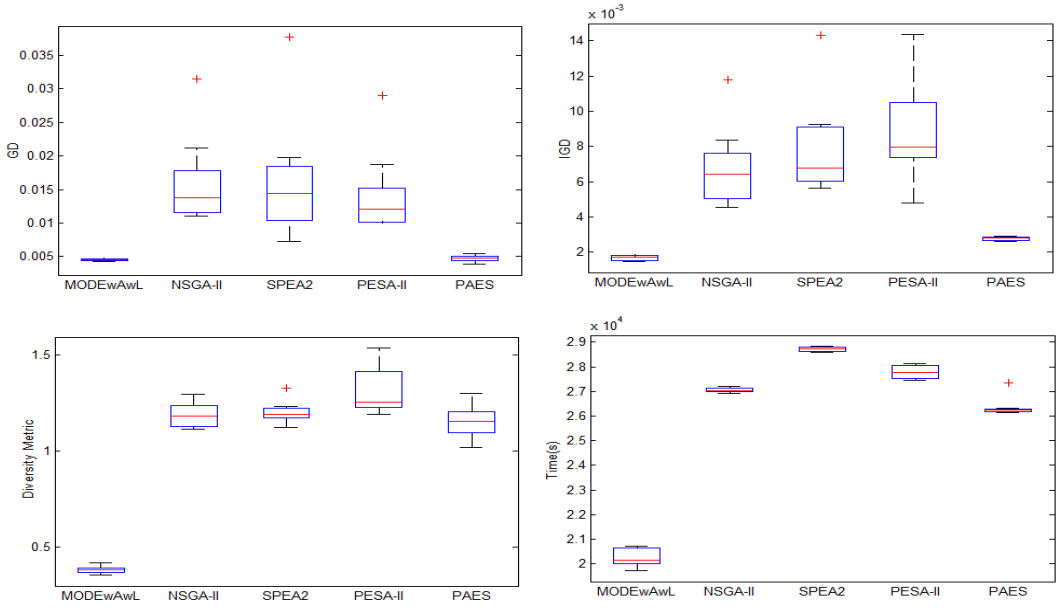


Figure A.7: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Russell 2000 dataset $\mathcal{K} = 5$.

A.2 Additional Experimental Test2

This section provides the additional computational results of five MOEAs considered in chapter 6 by the following parameter values.

Constraint values: $\mathcal{K} = 15$, $\epsilon_i = 0.01$, $\delta_i = 1.0$, $z = \{30\}$, $\vartheta_i = 0.008$

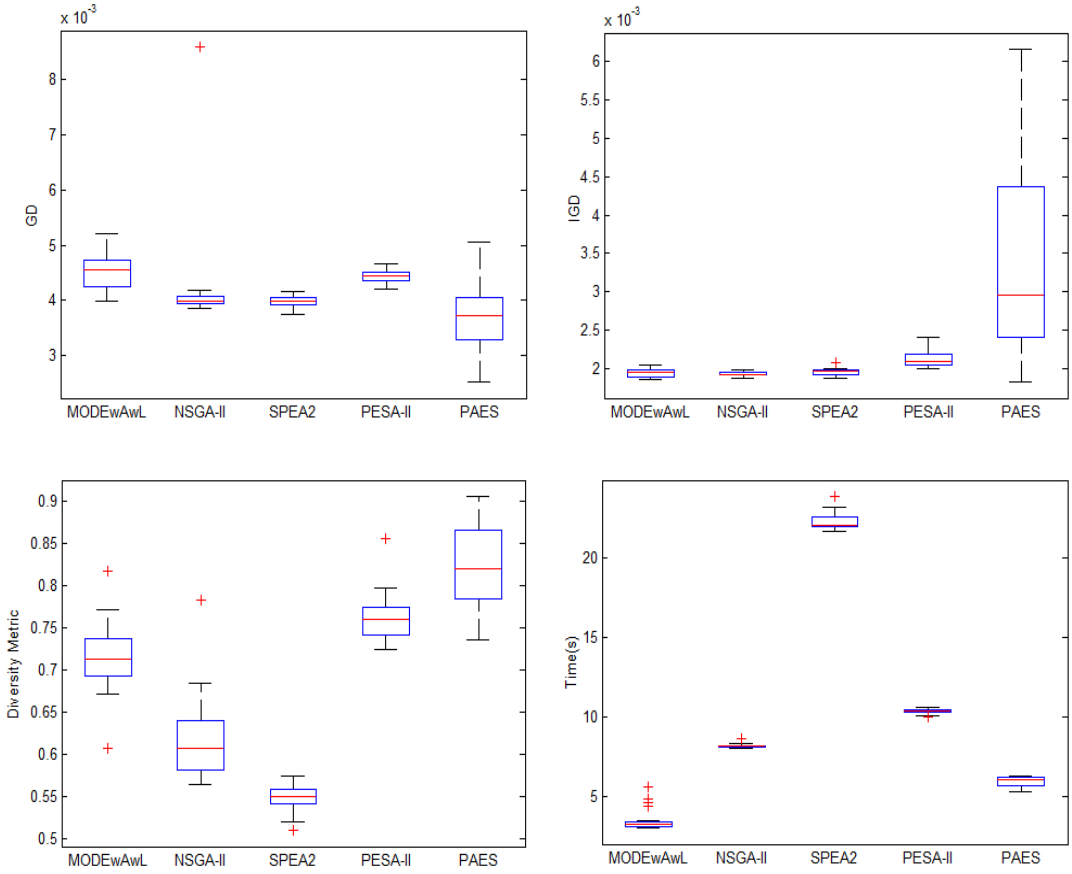


Figure A.8: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Hang Seng dataset $\mathcal{K} = 15$.

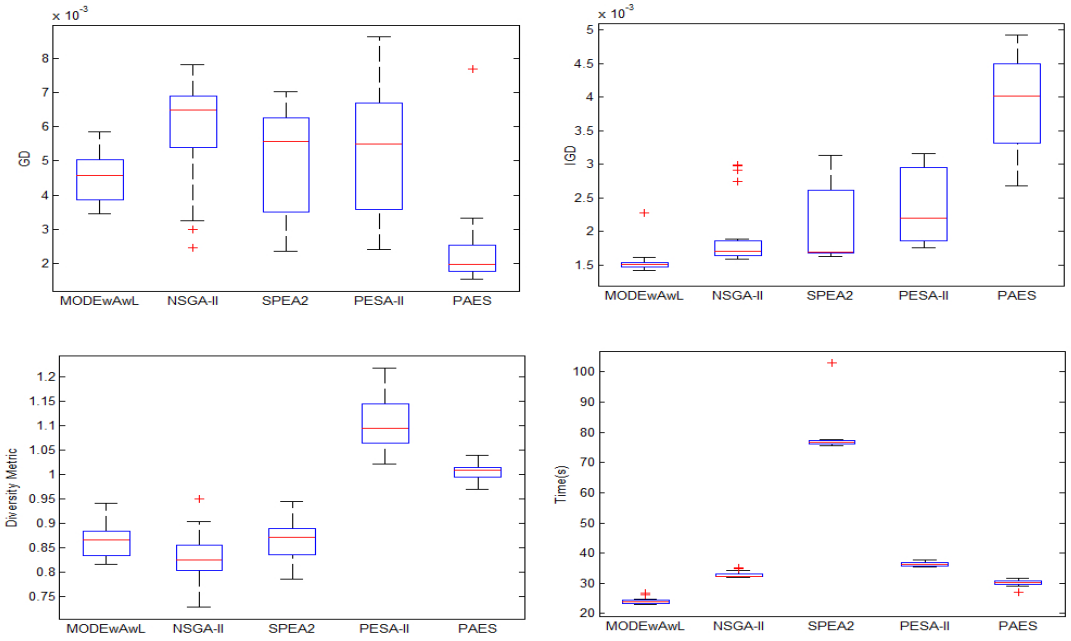


Figure A.9: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for DAX 100 dataset $\mathcal{K} = 15$.

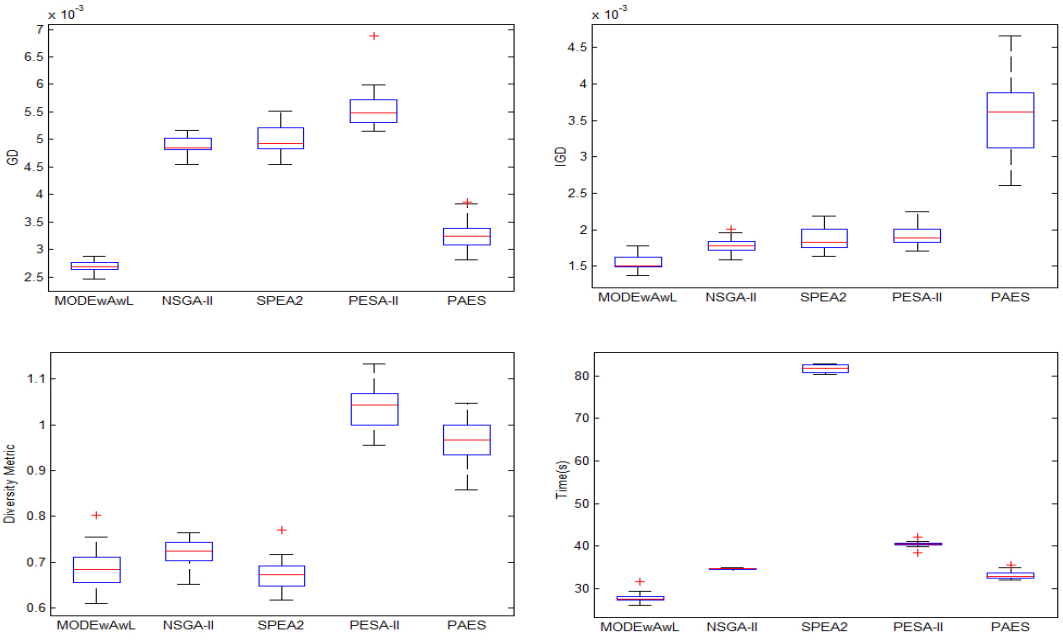


Figure A.10: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for FTSE 100 dataset $\mathcal{K} = 15$.

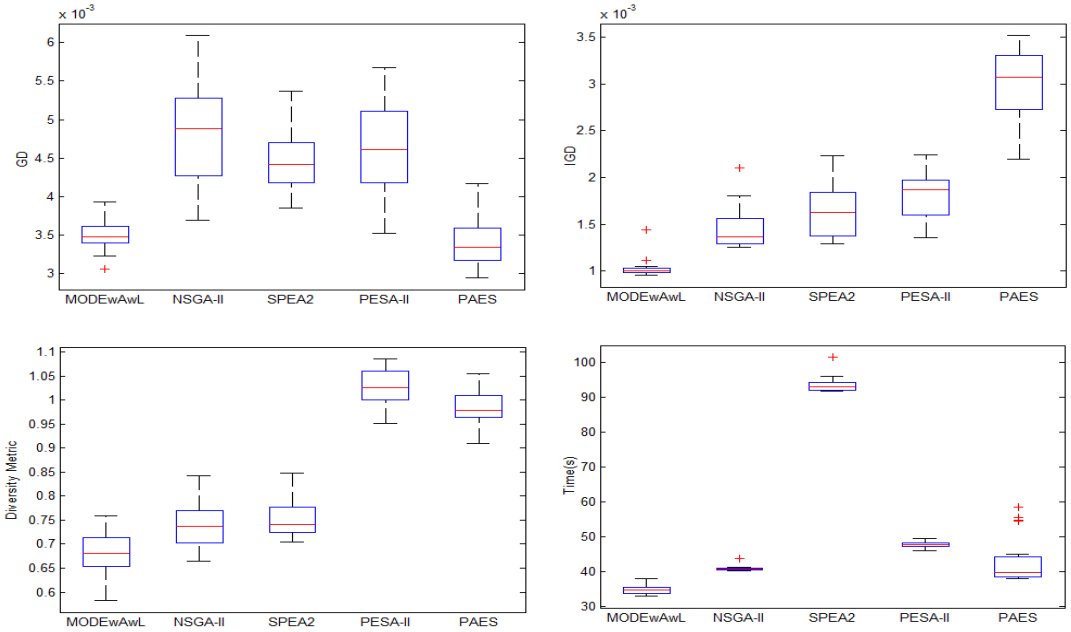


Figure A.11: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for S & P 100 dataset $\mathcal{K} = 15$.

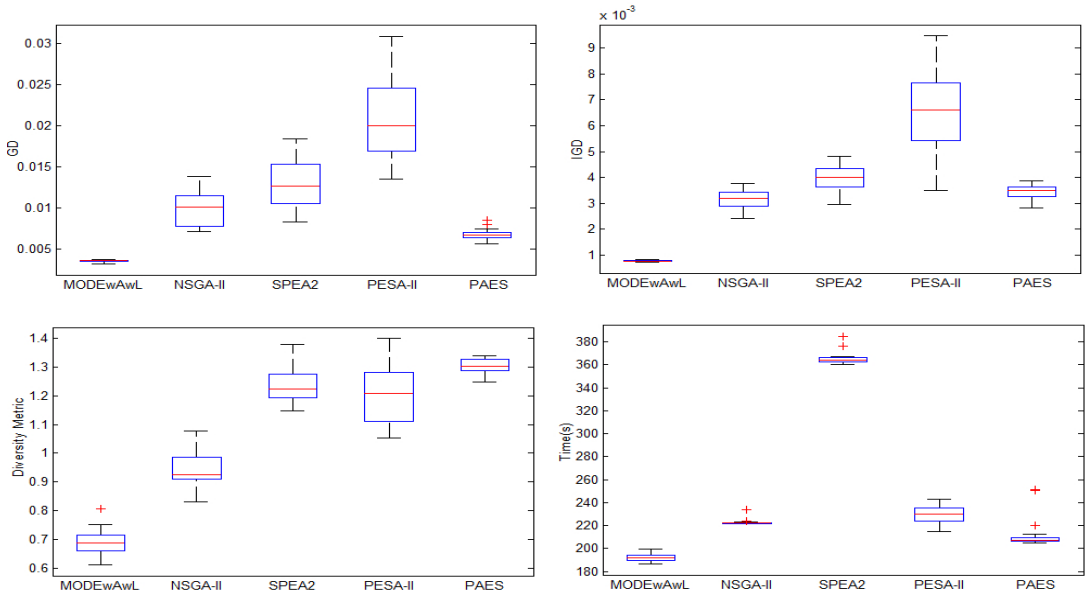


Figure A.12: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Nikkei dataset $\mathcal{K} = 15$.

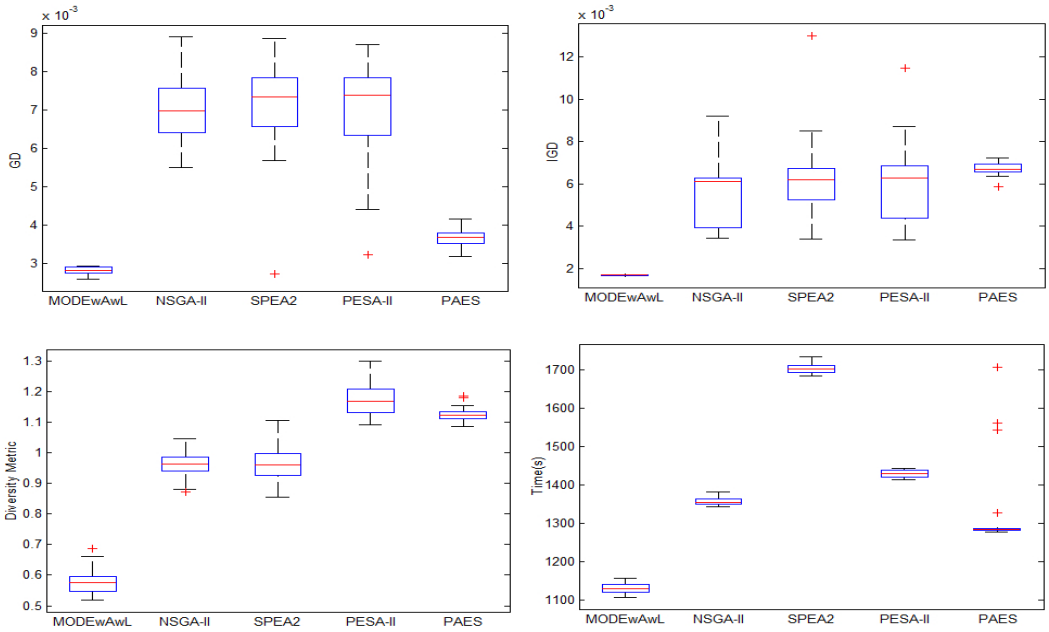


Figure A.13: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for S & P 500 dataset $\mathcal{K} = 15$.

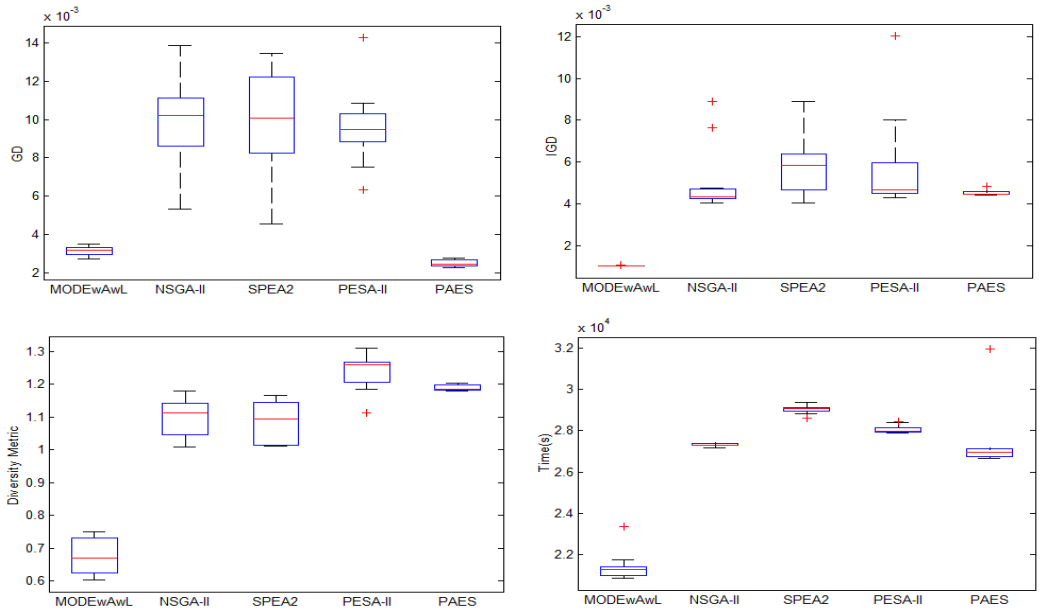


Figure A.14: Performance comparisons of five algorithms in terms of GD, IGD and Diversity (Δ) metrics for Russell 2000 dataset $\mathcal{K} = 15$.

Appendix B

B.1 OR-Library Dataset Example

In each dataset, the mean return of individual assets, standard deviation and the correlation between assets are given as shown in Table B.2. The covariance between two assets is evaluated from the correlation matrix as follows:

$$\sigma_{ij} := \rho_{ij} \times sd_i \times sd_j$$

where ρ_{ij} is the correlation value between the i^{th} and j^{th} assets, sd_i is the standard deviation of i^{th} asset and sd_j is the standard deviation of the j^{th} asset.

Asset	**Mean Return**	**Standard Deviation**	**Correlation Matrix**				
			1	**2**	**3**	**4**	**5**
1	.001309	.043208	1	.562289	.746125	.707857	.336386
2	.004177	.040258		1	.625215	.570407	.465845
3	.001487	.041342			1	.757165	.338075
4	.004515	.044896				1	.379100
5	.010865	.069105					1

Table B.2: Example data for first five assets of Hang Seng dataset (D1).

B.2 Example Dataset for mean-VaR Model

In non-parametric method, historical returns are used for the estimation of future returns. For the empirical part of this research, the daily adjusted close prices of all considered assets were gathered from the Yahoo! Finance web site[7]. These adjusted price series were used to calculate the log returns of the stocks as follows:

$$r_{it} = log\frac{\mathbb{p}_{i,t}}{\mathbb{p}_{i,t-1}}$$

where r_{it} denotes the return of the i^{th} asset at time t and $p_{i,t}$ denotes the price of the i^{th} asset at time t. An example dataset with three assets is shown in Table B.3.

Date	Asset Prices			Asset Returns		
	CoCa-Cola	3M	Halliburton	CoCa-Cola	3M	Halliburton
20/02/2008	58.23	79.94	36.72	0.002235	0.001126	0.01068
19/02/2008	58.10	79.85	36.33	-0.011296	-0.001252	0.013579
15/02/2008	58.76	79.95	35.84	0.003580	-0.004244	0
14/02/2008	58.55	80.29	35.84	-0.014245	-0.013484	0.004755
13/02/2008	59.39	81.38	35.67	-0.008884	0.028417	0.031324
⋮	⋮	⋮	⋮	⋮	⋮	⋮
07/03/2005	43.72	86.71	43.77	0.001832	0.002887	-0.023259
04/03/2005	43.64	86.46	44.80	0.010597	0.018208	0.016202
03/03/2005	43.18	84.9	44.08	0.003248	0.004013	0.007743
02/03/2005	43.04	84.56	43.74	-0.005330	0.001065	0.020557
01/03/2005	43.27	84.47	42.85	0.010921	0.006294	-0.025802

Table B.3: Example of daily financial time series data for three assets over a period of 750 trading days.

[7] http://finance.yahoo.com

B.3 Constituents of DS1 and DS2 datasets

Each stock has a unique ticker symbol representing a particular security traded on a particular stock market. For example, AAPL is a unique ticker symbol for stock Apple Inc. The unique ticker symbols of the constituents of datasets DS1 and DS2 are provided in Table B.4 and Table B.5 respectively.

APPL	BA	COP	EMR	HD	MCD	NOV	SO	VZ
ABT	BAC	COST	EXC	HON	MDLZ	NSC	SPG	WAG
ACN	BAX	CSCO	F	HPQ	MDT	ORCL	T	WFC
AEP	BK	CVS	FCX	IBM	MET	OXY	TGT	WMB
AIG	BMY	CVX	FDX	INTC	MMM	PEP	TWX	WMT
ALL	BRK.B	DD	FOXA	JNJ	MO	PFE	TXN	XOM
AMGN	C	DIS	GD	JPM	MON	PG	UNH	
AMZN	CAT	DOW	GE	KO	MRK	QCOM	UNP	
APA	CL	DVN	GILD	LLY	MS	RTN	UPS	
APC	CMCSA	EBAY	GS	LMT	MSFT	SBUX	USB	
AXP	COF	EMC	HAL	LOW	NKE	SLB	UTX	

Table B.4: List of 94 Securities of S & P 100

A	BDX	CSCO	EXPD	HP	LRCX	NUE	RL	TWC
AA	BEN	CSX	EXPE	HPQ	LSI	NVDA	ROK	TWX
AAPL	BHI	CTAS	F	HRB	LUK	NWL	ROP	TXN
ABC	BIIB	CTL	FAST	HRL	LUV	OI	ROST	TXT
ABT	BK	CTSH	FCX	HRS	M	OKE	RRC	TYC
ACE	BLK	CTXS	FDO	HSP	MA	OMC	RSG	UA
ACN	BLL	CVC	FDX	HST	MAC	ORCL	RTN	UNH
ACT	BMS	CVS	FE	HSY	MAR	ORLY	SBUX	UNM
ADBE	BMY	CVX	FFIV	HUM	MAS	OXY	SCG	UNP
ADI	BRCM	D	FIS	IBM	MAT	PAYX	SCHW	UPS
ADM	BSX	DAL	FISV	ICE	MCD	PBCT	SE	URBN
ADP	BTU	DD	FITB	IFF	MCHP	PBI	SEE	USB
ADS	BWA	DE	FLIR	IGT	MCK	PCAR	SHW	UTX
ADSK	BXP	DFS	FLR	INTC	MCO	PCG	SIAL	V
AEE	C	DGX	FLS	INTU	MDLZ	PCL	SJM	VAR
AEP	CA	DHI	FMC	IP	MDT	PCLN	SLB	VFC
AES	CAG	DHR	FOSL	IPG	MET	PCP	SNA	VIAB
AET	CAH	DIS	FOXA	IR	MHFI	PDCO	SNDK	VLO
AFL	CAM	DISCA	FRX	IRM	MHK	PEG	SNI	VMC
AGN	CAT	DLTR	FSLR	ISRG	MKC	PEP	SO	VNO
AIG	CB	DNB	FTI	ITW	MMC	PETM	SPG	VRSN
AIV	CBG	DNR	FTR	IVZ	MMM	PFE	SPLS	VRTX
AIZ	CBS	DO	GAS	JBL	MNST	PFG	SRCL	VTR
AKAM	CCE	DOV	GCI	JCI	MO	PG	SRE	VZ
ALL	CCI	DOW	GD	JEC	MON	PGR	STI	WAG
ALTR	CCL	DPS	GE	JNJ	MOS	PH	STJ	WAT
ALXN	CELG	DRI	GGP	JNPR	MRK	PHM	STT	WDC
AMAT	CERN	DTE	GHC	JOY	MRO	PKI	STX	WEC
AME	CF	DTV	GILD	JPM	MS	PLD	STZ	WFC
AMGN	CHK	DUK	GIS	JWN	MSFT	PLL	SWK	WFM
AMP	CHRW	DVA	GLW	K	MSI	PM	SWN	WHR
AMT	CI	DVN	GMCR	KEY	MTB	PNC	SWY	WIN
AMZN	CINF	EA	GME	KIM	MU	PNR	SYK	WLP
AN	CL	EBAY	GNW	KLAC	MUR	PNW	SYMC	WM
AON	CLX	ECL	GOOGL	KMB	MWV	POM	SYY	WMB
APA	CMA	ED	GPC	KMX	MYL	PPG	T	WMT
APC	CMCSA	EFX	GPS	KO	NBL	PPL	TAP	WU
APD	CME	EIX	GRMN	KR	NBR	PRGO	TDC	WY
APH	CMG	EL	GT	KSS	NDAQ	PRU	TE	WYN
ARG	CMI	EMC	GWW	KSU	NE	PSA	TEG	WYNN
ATI	CMS	EMN	HAL	L	NEE	PVH	TEL	X
AVB	CNP	EMR	HAR	LB	NEM	PWR	TGT	XEL
AVP	CNX	EOG	HAS	LEG	NFLX	PX	THC	XL
AVY	COF	EQR	HBAN	LEN	NFX	PXD	TIF	XLNX
AXP	COG	EQT	HCBK	LH	NI	QCOM	TJX	XOM
AZO	COH	ESRX	HCN	LLL	NKE	R	TMK	XRAY
BA	COL	ESS	HCP	LLTC	NOC	RAI	TMO	XRX
BAC	COP	ESV	HD	LLY	NOV	RDC	TROW	YHOO
BAX	COST	ETFC	HES	LM	NRG	REGN	TRV	YUM
BBBY	COV	ETN	HIG	LMT	NSC	RF	TSCO	ZION
BBT	CPB	ETR	HOG	LNC	NTAP	RHI	TSN	ZMH
BBY	CRM	EW	HON	LO	NTRS	RHT	TSO	
BCR	CSC	EXC	HOT	LOW	NU	RIG	TSS	

Table B.5: List of 475 Securities of S & P 500