EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION IN INVESTMENT PORTFOLIO MANAGEMENT

CHIAM SWEE CHIANG
(B.Eng (Hons.), NUS)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2009

Abstract

Many real-world problems involve the simultaneous optimization of several competing objectives and constraints that are difficult, if not impossible, to solve without the aid of powerful optimization algorithms. As no one solution is optimal to all objective in the presence of conflicting specifications, the optimization algorithms must be capable of generating a set of alternative solutions, representing the tradeoffs between the objectives. Evolutionary algorithms, a class of population-based stochastic search technique, have shown general success in solving complex real-world multi-objective optimization problems, where conventional optimization tools failed to work well. Its main advantage lies in its capability to sample multiple candidate solutions simultaneously, hence enabling the entire set of Pareto-optimal solutions to be approximated in a single algorithmic run. Much work has been devoted to the development of multi-objective evolutionary algorithms in the past decade and it is increasingly finding application to the diverse fields of engineering, bioinformatics, logistics, economics, finance, and etc.

This thesis focuses particularly on investment portfolio management, an important subject in the field of economics and finance, where the central theme is the professional management of an appropriate mix of financial assets to satisfy specific investment goals. The decision process will typically involve issues such as asset allocation, security selection, performance measurement, management styles and etc. Due to the complexity of these issues, classical optimization tools from the realm of operations research are restricted to a limited set of problems and/or the optimization models have to accept strong simplifications. These restrictions have thus motivated the development and application of evolutionary optimization techniques for this purpose. As such, the primary motivation of this thesis is to provide a comprehensive treatment on the design and application of multi-objective evolutionary algorithms to address the several key issues involved with investment portfolio management, namely asset allocation and portfolio management style.

For asset allocation, the mean-variance model developed by Harry Markowitz, widely regarded as the foundation of modern portfolio theory, is considered to provide the quantitative framework for this optimization problem. A generic multi-objective evolutionary algorithm designed specifically for portfolio optimization is proposed and its feasibility is evaluated based on a rudimentary instantiation of the mean-variance model. Avenues to incorporate user preferences into the portfolio construction process are examined also. In addition, real-world constraints arising from business/industry regulations and practical concerns are incorporated to enhance the realism of the mean-variance model and the impacts on the efficient frontier are studied.

The second part of this work is concerned with portfolio management style, which can be broadly classified as active and passive. While active management relies on the belief that excess yields over market average are attainable by exploiting market inefficiencies, passive management centers on efficient financial markets and aims to replicate returns-risk profiles similar to market indices. For the former, security selection through technical analysis is studied, where a multi-objective evolutionary platform is developed to optimize technical trading strategies capable of yielding high returns at minimal risk. Popular technical indicators used commonly in real-world practices are used as the building blocks for these strategies, which hence allow the examination of their trading characteristics and behaviors on the evolutionary platform. In the aspect of passive management, a realistic instantiation of the index tracking optimization problem that accounted for stochastic capital injections, practical transactions cost structures and other real-world constraints is formulated and used to evaluate the feasibility of the proposed multi-objective evolutionary platform that simultaneously optimized tracking performance and transaction costs throughout the investment horizon.

Acknowledgements

First and foremost, I will like to thank my thesis supervisor, Professor Tan Kay Chen for introducing me to the wonderful field of computational intelligence and his continuous support and guidance throughout my course of study. His understanding, encouragments and personal guidances provided the basis for this thesis. I will also like to thank my co-supervisor, Professor Abdullah Al Mamun for his important support throughout this work.

All my lab buddies at the Control and Simulation laboratory made it a convivial place to work. In particular (in order of seniority), I will like to TAC-Q Chi Keong for showing me the way of research, Dasheng for his invaluable contributions to the research group, Eujin who accompanied me to the world of finance, Brian AND Chun Yew for the soap and drama, Hanyang for keeping me on course and Chin Hiong for his tips! Jokes aside, this bunch of great folks, as well all others in C&S lab, have inspired me in research and life through our interactions and stimulating discussions during the long hours in the lab. Thanks!

I owe my loving thanks to my wife Pricilla, who has been extremely kind and understanding during this period of my life. Without her encouragement and understanding, it would have been impossible for me to finish this work. Also, my special gratitude is due to my entire family, notably my two sisters Valerie and Siew Sze for providing me a loving environment.

Lastly and most importantly, I wish to thank my parents, Tony and Judy. They bore me, raised me, supported me, taught me, and loved me. To them, I dedicate this thesis.

Publications

1. S. C. Chiam, K. C. Tan and A. A. Mamun, "A Memetic Model of Evolutionary PSO for Computational Finance Applications," *Expert Systems With Applications*, vol. 36, no. 2, pp. 3695-3711, 2009.

2. S. C. Chiam, K. C. Tan and A. A. Mamun, "Investigating technical trading strategy via an multi-objective evolutionary platform," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10408-10423, 2009.

3. K. C. Tan, S. C. Chiam, A. A. Mamun and C. K. Goh, "Balancing Exploration and Exploitation with Adaptive Variation for Evolutionary Multi-objective Optimization," *European Journal of Operational Research*, vol. 197, no. 2, pp. 701-713, 2009.

4. S. C. Chiam, K. C. Tan, C. K. Goh and A. A. Mamun, "Improving Locality in Binary Representation via Redundancy," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 3, pp. 808-825, 2008.

5. S. C. Chiam, K. C. Tan and A. A. Mamun, "Evolutionary multi-objective portfolio optimization in practical context," *International Journal of Automation and Computing*, vol. 5, no. 1, pp. 67-80, 2008.

6. S. C. Chiam, K. C. Tan and A. A. Mamun, "Molecular Dynamics Optimizer," *Fourth International Conference on Evolutionary Multi-Criterion Optimization*, Matsushima, Japan, March 5-8, pp. 302-316, 2007.

7. S. C. Chiam, K. C. Tan and A. A. Mamun, "Multiobjective Evolutionary Neural Networks for Time Series Forecasting," *Fourth International Conference on Evolutionary Multi-Criterion Optimization*, Matsushima, Japan, March 5-8, pp. 346-360, 2007.

8. S. C. Chiam, C. K. Goh and K. C. Tan, "Adequacy of Empirical Performance Assessment for Multiobjective Evolutionary Optimizer," *Fourth International Conference on Evolutionary Multi-Criterion Optimization*, Matsushima, Japan, March 5-8, pp. 893-907, 2007.

9. C. Y. Cheong, S. C. Chiam and C. K. Goh, "Eliminating Positional Dependency in Binary Representation via Redundancy," *2007 IEEE Symposium on Foundations of Computational Intelligence Article*, Honolulu 1-5 April, pp. 251-258, 2007.

10. S. C. Chiam, K. C. Tan, A. A. Mamun and Y. L. Low, "A Realistic Approach to Evolutionary Multi-objective Portfolio Optimization," *IEEE Congress on Evolutionary Computation 2007*, Singapore, September 25-28, pp. 204-211, 2007.

11. S. C. Chiam, C. K. Goh and K. C. Tan, "Issues of Binary Representation in Evolutionary Algorithms, " *The 2nd IEEE International Conference on Cybernetics & Intelligent Systems*, Bangkok, Thailand, June 7-9, 2006.

12. C. K. Goh, S. C. Chiam and K. C. Tan, "An Investigation on Noisy Environments in Evolutionary Multi-Objective Optimization, " *The 2nd IEEE International Conference on Cybernetics & Intelligent Systems*, Bangkok, Thailand, June 7-9, 2006.

13. E. J. Teoh, S. C. Chiam, C. K. Goh and K. C. Tan, "Adapting evolutionary dynamics of variation for multi-objective optimization," *IEEE Congress on Evolutionary Computation 2005*, Edinburgh, UK, vol. 2, pp. 1290-1297,2005.

14. C. K. Goh, K. C. Tan, D. S. Liu, S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *European Journal of Operational Research*, accepted.

Contents

List of Figures

List of Tables

Chapter 1

Investment Portfolio Management

Investment portfolio management is the professional management of an appropriate mix of financial assets to meet specific investment goals for the benefit of the investors. In modern financial markets, there is a huge variety of asset classes in which one may invest their wealth. They broadly range from traditional financial products like stocks, bonds, money markets and cash to alternative investments like commodities, financial derivatives, hedge funds, real estate, private equity, as well as venture capital. While some are standardized products that are publicly quoted and traded on exchanges, others are specially engineered to cater for specific needs of the investor and are traded over the counter, hence associated with lower liquidity.

Faced with an extensive range of financial assets with distinct characteristics, the crux of the problem lies in finding the optimal portfolio mix to meet investor needs. The optimization process will involve issues like asset allocation, security selection, performance measurement, management styles and etc. and the main objective in investment portfolio management is to deliver solutions for these issues. Without any loss in generality, this work will specifically focus on asset allocation and management styles. For brevity, discussions and empirical analyzes will be restricted to equity portfolios though the generality of the proposed solution techniques allows extensibility to other asset classes.

The problem of asset allocation focuses on the allocation of fund to each portfolio asset so as to maximize the (expected) consumption utility during a specified investment period and/or total wealth at the end of the investment period. The first half of this chapter will present a brief overview of the mean-variance model pioneered by Harry Markowitz [166] (one of the earliest and prominent work in the field of investment portfolio management) and highlight some of its limitations. This model will be used subsequently in this work as the quantitative framework for the asset allocation problem.

Portfolio management style can be broadly classified as active and passive. While active management relies on the belief that excess yields over market average are attainable by exploiting market inefficiencies, passive management centers on efficient financial markets and aims to replicate returns-risk profiles similar to market indices. The second half of this chapter will provide a general introduction to these styles and highlight some of their key differences.

1.1 Asset Allocation via Mean-Variance Analysis

Asset allocation is one of the crucial steps in investment portfolio management, which determines the proportion of fund to be invested in each portfolio constituent. Security selection will then follow where the appropriate securities for each portfolio subset are being determined. Following that, asset allocation will be triggered again to determine the appropriate mix in each portfolio subset. While various approaches (like tactical asset allocation, insured asset allocation and etc) exist for this purpose, this work will specifically focus on mean-variance analysis.

1.1.1 Mean-Variance Model

Portfolio management entails elements of stochastic optimization as returns from financial instruments are probabilistic in nature. Figure 1.1 compares the daily stock prices of DBS and UOB (the two largest local banks in Singapore) from the period 01012008 to 05082008. Clearly, it is impossible to foretell the future price movement of each stock as their relative performance varied with time. As such, performance measurement and analysis of financial products and investment portfolios as a whole should involve statistical and/or probabilistic elements. One trivial approach is to describe individual stock price returns via probability distribution and measure portfolio performance with the aggregate expected returns. However, asset allocation based on this naive objective will led to unreasonable portfolio choices [19].

To investigate this further, consider an investor allocating all of his wealth to N different assets indexed by $i = 1, ..., N$ and the returns of each asset is a random variable r_i with expected value

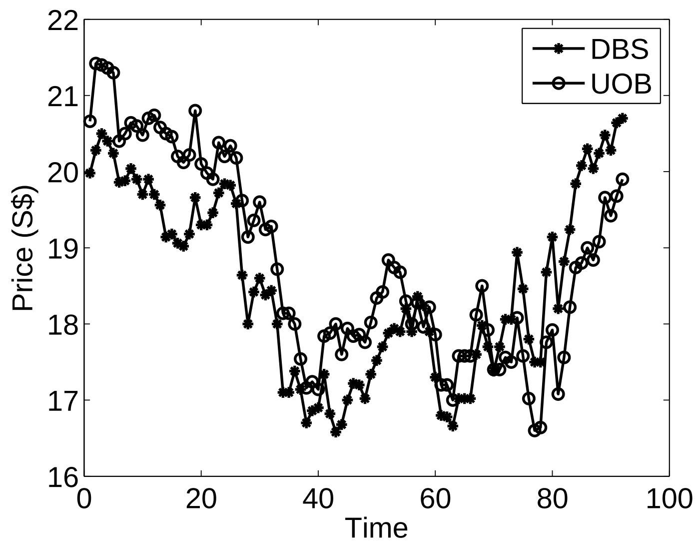


Figure 1.1: Daily price series of DBS and UOB (i.e. the two largest bank stocks in terms of capitalization value in the Straits Times Index, Singapore) for the period between 01012008 and 05082008.

$\mu_i = E(r_i)$. The fraction of wealth invested in asset i is represented by a decision variable w_i which is bounded by $0 \leq w_i \leq 1$, assuming short-selling is dis-allowed. The expected portfolio returns will be simply as follows:

$$E(\sum_{i=1}^{n} r_i w_i) = \sum_{i=1}^{n} E(r_i) w_i = \sum_{i=1}^{n} \mu_i w_i \tag{1.1}$$

Consequently, The asset allocation problem can be formalized as follows:

$$\max \sum_{i=1}^{n} \mu_i w_i \tag{1.2}$$

$$\text{s.t. } \sum_{i=1}^{n} w_i = 1, w_i \geq 0$$

The solution to (1.2) is rather trivial, as it simply involves choosing the stock with the maximum expected return, $i^* = arg \max_{i=1,\ldots,n} \mu_i$ and setting $w_{i^*} = 1$. Intuitively, this portfolio is analogous to putting all of the eggs in one single basket which is extremely risky. This simple example highlights the importance of supplementing the expected returns with other information. A natural extension in this context is the incorporation of risk measures and the most common measure for this purpose is the standard deviation of the expected returns. Typically, higher risk is associated with greater dispersion of returns around the expected value, as it translates to greater uncertainty of future returns.

The asset universe in (1.2) is reduced to the two assets in Figure 1.1. Their expected returns, $E(r_{DBS})$ and $E(r_{UOB})$ are 0.60% and -0.20% respectively (based on the historical price series) and their standard deviation (σ_{DBS} and σ_{DBS}) are 0.40% and 0.42% respectively. On face value, it can be concluded that the investor should allocate all his wealth to DBS due to it having a higher expected returns and lower standard deviation, which corresponds to lower risk. However, this analysis is not complete, as the correlation between them has been neglected. Including UOB can offer diversification benefits, if their returns are not highly correlated in a positive sense i.e. if DBS performs poorly, UOB might perform well to mitigate the loss.

The variance for this two-asset portfolio is as follows:

$$\begin{aligned} \sigma^2 &= Var(w_{DBS}r_{DBS} + w_{UOB}r_{UOB}) \\ &= w_{DBS}^2\sigma_{DBS}^2 + w_{UOB}^2\sigma_{UOB}^2 + 2w_{DBS}w_{UOB}\sigma_{DBS,UOB} \end{aligned} \tag{1.3}$$

where w_{DBS} and w_{UOB} represent the proportion of wealth invested in DBS and UOB respectively and $\sigma_{DBS,UOB}$ denotes the covariance between them. Generalizing this relationship for larger problem sets, the equation for variance is as follows:

$$\sigma^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{i,j} \tag{1.4}$$

where $\sigma_{i,j}$ denotes the covariance between asset i and j.

Different weight combinations will correspond to different portfolios characterized by their expected returns and standard deviation. Ideally, an investor will want to maximize the expected return and minimize returns variance. This principle forms the fundamentals of the famous Markowitz mean-variance model [166], where the problem can be formulated as such,

- For a given upper bound of σ^2 for the variance of the portfolio return, find an admissible portfolio π^* such that $\mu(\pi^*)$ is maximal under all admissible portfolio π, with $\sigma^2(\pi^*) \leq \sigma^2$.

- For a given lower bound of μ for the mean of the portfolio return, find an admissible portfolio π^* such that $\sigma^2(\pi^*)$ is minimal under all admissible portfolio π, with $\mu(\pi^*) \geq \mu$.

Applying the mean-variance analysis for the two-stocks (i.e. DBS and UOB) asset allocation problem, different weight combination were considered and the risk-return profile of the different portfolios are plotted in Figure 1.2. As the two objectives are inherently conflicting in nature, the optimum solutions will essentially comprise of a set of solution illustrating the trade-off between them. Intuitively, an investor will want the highest return for a given level of risk. As such, only the upper bound of the plot will be considered which is commonly known as the efficient frontier.

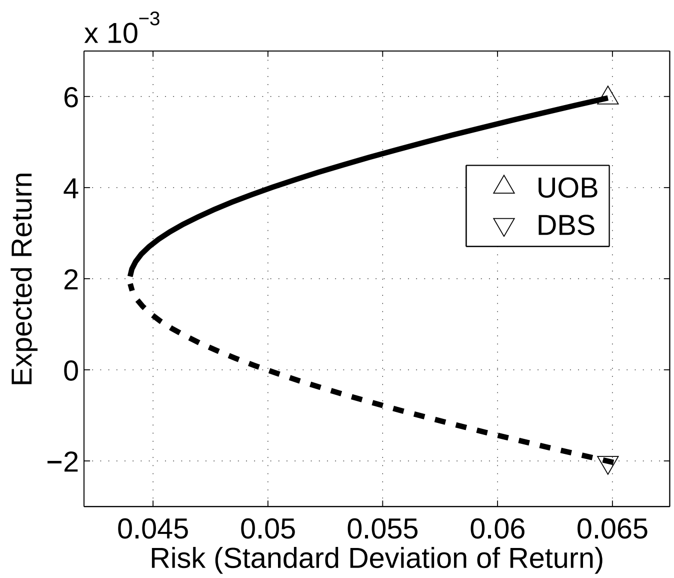


Figure 1.2: Plot showing the risk-return profiles by considering different weights combinations in the two-asset (i.e. DBS and UOB) portfolio optimization problem. Efficient frontier is highlighted in bold.

1.1.2 Limitations of Markowitz Model

The essence of mean-variance analysis is to construct portfolios amongst the pool of assets available, offering the highest expected returns and lowest risk possible. In this single-period decision problem, a one-off decision will be made at the beginning of the investment period with no further actions thereafter. The aim is to maximize the terminal wealth at the end of the period. Till date, this model still holds great importance in real-world applications and is widely considered in financial institutions.

Despite its prominent role in financial theory, the mean-variance model has constantly been the subject of widespread criticism. For example, the fundamental assumption of a perfect market without taxes and transactions costs, where securities are infinitely divisible and therefore can be traded in any fraction, is highly unrealistic in practical context. Also, the normality assumption in returns distribution contradicts the rather well-known observations that empirical distribution of asset returns exhibit non-symmetry and excess kurtosis. The direct implication is that the first two moments of expected return and variance are insufficient to describe the portfolio fully and higher moments are required. These limitations have consequently motivated further development to improve its realism and relevance to the asset allocation problem in real-world.

Related literatures have extended the mean-variance model by modifying the existing objective functions. Particularly, Arnone et al. [8] and Loraschi et al. [154] considered downside risk (i.e. distribution of the downside returns) in place of the returns volatility. Alternatively, additional objective functions have been incorporated to enhance the original model. For example, Fieldsend et al. [82] considered the portfolio size as an additional objective to be optimized, allowing the 2-dimensional cardinality constrained frontier for any particular cardinality to be obtained directly. Other objectives considered in literature included surplus variance [229], portfolio Value-at-Risk [229], annual dividend [72] and asset ranking [72]. Also, as portfolio managers often face a number of realistic constraints arising from pre-specified investment mandates, business/industrial regulations and other practical issues [221], these constraints have been incorporated into mean-variance model in related works.

Being cast in a single-period framework, the mean-variance model essentially represents a passive buy-and-hold strategy that remained indifferent to the ever-changing market conditions. Clearly, it is counter-intuitive to assume a static relationship for the different assets in the portfolio. For example, correlation will typically rise in stock market crashes, just when diversification is most needed. This has thus motivated the consideration of the dynamic nature of investment portfolio management with the work of Merton [169, 170] being widely regarded as the real starting point in the field of continuous-time portfolio management. Hybrid variants like multi-period portfolio management also exist where the investment horizon is split into discrete time periods and the mean-variance criteria are considered in every period. Portfolio rebalancing strategies and transaction costs are important considerations in multi-period and dynamic portfolio management.

1.2 Investment Portfolio Management Styles

Portfolio management styles could be broadly classified into passive or active. While active management relies on the belief that excess yields over market average are attainable by exploiting market inefficiencies, passive management centers on efficient financial markets and aims to replicate similar returns-risk profiles as market indices. There is an inherent tradeoff between these two styles, i.e. the low-cost but less-exciting alternative of passive investing versus the higher-cost but potentially more lucrative alternative of active investing [197].

Before reviewing them in detail, it is imperative to introduce the efficient market hypothesis [78]. Essentially, this hypothesis asserts that financial markets are "informationally efficient ", or that price on traded financial assets already reflect all known information and therefore are unbiased in the sense that they reflect the collective beliefs of all investors about future prospects. As such, it is not possible to consistently outperform the market by using any information that the market already knows. Information or news here denotes anything that may affect prices, is unknowable in the present and thus appears randomly in the future.

1.2.1 Active Portfolio Management

Active portfolio management is an attempt by the manager to make specific investments with the goal of outperforming a pre-determined benchmark index, net of transaction costs, on a risk-adjusted basis. The central belief is that the financial markets are not efficient and such opportunities can be exploited for profits. As such, mangers are essentially "betting "against markets being perfectly efficient and these "bets "can be broadly categorized into fundamental and technical.

The realm of fundamental analysis is one where mispricing might temporarily exist in the short term before market forces rectified this pricing discrepancy in the long run. Fundamentalists will analyze the market forces of demand and supply to determine the intrinsic value of financial assets and enter (exit) the market if it is below (above) its intrinsic value, which is a sign of undervaluation (overvaluation). The unit of interest here could be a particular security name, where its market price is compared against the valuation implied from financial statement analysis discounted cash flow model, or escalate to asset class level, where the relative value between the various asset classes are assessed. It can also be based on specific sector classification like industrial (manufacturing, construction, finance), product (consumer, industrial, services), perceived characteristics (growth, cyclical, stable) and etc.

In stark contrast, technicians completely ignore market fundamentals and decide solely based on market action i.e. the past history of market prices and trading action. The central idea is that all available information is already reflected in the market prices, hence rendering the usefulness of fundamental analysis. Through the extrapolation of historical price patterns, technicians assume either past stock price trends will continue in the same direction or they will reverse themselves.

In the context of investment portfolio management, active management views can be reflected in asset allocation where for example the portfolio weights for undervalued securities are temporarily increased at the expense of overvalued securities, until the abnormalities have been rectified. These views can be considered in security selection where for example, technical indicators being used to limit the entire stock universe to a manageable list of "potential "names.

Clearly, the effectiveness of an actively-managed investment portfolio obviously depends on the skill of the manager. In reality, the majority of active mangers rarely outperform their index counterparts over long periods of time, for example, the Standard & Poor's Index Versus Active scorecards demonstrate that only a minority of actively managed mutual funds usually beat Standard & Poor's various index benchmarks. In fact, this minority percentage tends to shrink further as the comparison period lengthens. Accounting for all expenses, underperformance is possible even if the portfolio outperforms the market. Nevertheless, active management remains an attractive strategy within market segments that are less likely to be fully efficient, such as investments in small cap stocks.

1.2.2 Passive Portfolio Management

While active management relies on the belief that excess yields over market average are attainable by exploiting market inefficiencies, passive management centers on efficient financial markets and aims to replicate similar returns-risk profiles as market indices. The implicit assumption here is that financial markets are efficient and no financial strategy can regularly outperform the market average. As such, the objective here is to generate market returns by replicating financial indexes as best as possible. There exist many different types of indexes for various broad market categories, for example equity indexes (S&P 500 & Nasdaq composite index), indexes for small capitalization stocks (Russell 2000) and value/growth oriented stocks (Russell Value/Growth index), indexes for world regions (MSCI World), as well as for smaller regions, individual countries and the type of countries (emerging Asia markets). There exist also customized passive portfolios, known as completeness fund, that are constructed to complement active portfolios that do not cover the entire market. Instead of the published indexes highlighted, these funds will track customized indexes that incorporate the characteristics of stocks not covered by the active managers.

Even though passive portfolio management has a straightforward goal of matching the portfolio returns with respect to an underlying index, uncontrollable factors like cash inflows/outflows, company mergers and bankruptcies and etc, will translate to inevitable discrepancies in returns over time. While index funds generally aim to minimize turnover and transaction fees, rebalancing is undoubtedly essential also to prevent their returns from lagging the underlying index in the long run. This subject will be discussed in further detail later in Chapter 7.

1.3 Thesis Overview

The central theme in investment portfolio management is to manage an appropriate mix of financial assets to satisfy certain specified investment goals and this process requires portfolio managers to address various issues like asset allocation, security selection, performance measurement, management styles and etc. Many of these issues have been formulated as optimization problems and are widely studied in literature works. However, due to their sheer complexity, classical optimization tools from the realm of operations research were restricted to a limited set of problems and/or the corresponding optimization models had to accept strong simplifications. These restrictions have thus motivated the development and application of evolutionary optimization techniques for this purpose, as they have shown general success in solving complex real-world optimization problems from the diverse fields of engineering, bioinformatics, logistics, economics, finance, and etc. The primary motivation of this work is to provide a comprehensive treatment on the design and application of multi-objective evolutionary algorithms to address the issues involved with investment portfolio management, particularly asset allocation and management styles.

This thesis is organized as such. The first two chapters will provide the necessary background information on the subject matters. This chapter in particular focused on investment portfolio management and highlighted some of the associated issues that will be investigated further in subsequent chapters. Chapter 2 will continue with a brief overview on the key concepts and design issues involved with evolutionary multi-objective optimization, as well as a short introduction on memetic algorithms, a synergetic combination of global and local search strategies that corresponds to an effective and efficient optimization model.

Following that, Chapter 3 will formally introduce multi-objective evolutionary algorithm as the optimization platform for investment portfolio management. Specifically, this chapter will examine how the chromosomal representation and variation operations of evolutionary optimizers can be extended for the purpose of portfolio optimization and how algorithmic performance can be further enhanced via local search strategies and dynamism operators.

The rest of this thesis is divided into two main parts, with each part focusing on different aspect of investment portfolio management. The first part, comprising of Chapter 4 and 5, will focus on asset allocation. Specifically, the mean-variance model developed by Harry Markowitz, which is widely regarded as the foundation of modern portfolio theory, will be considered here to provide a quantitative framework for the asset allocation problem. Chapter 4 will evaluate the feasibility of the proposed multi-objective evolutionary platform based on a rudimentary instantiation of the mean-variance model and examine avenues to incorporate preferences into the decision-making process via a memetic model. In Chapter 5, real-world constraints arising from business/industry regulations and practical concerns will be incorporated to improve the realism of the mean-variance model and their impacts on the efficient frontier will be studied.

The second part of this thesis is concerned with the two distinct portfolio management styles. Active management, specifically technical analysis in the context of security selection, will be considered in Chapter 6. A multi-objective evolutionary platform that optimizes technical trading strategies capable of yielding high returns at minimal risk will be proposed. Popular technical indicators used commonly in real-world practices will be used as the building blocks for these strategies, hence allowing the examination of their trading characteristics and behaviors on the multi-objective evolutionary platform. Subsequently, Chapter 7 will switch to passive management, where a realistic instantiation of the index tracking optimization problem that accounted for stochastic capital injections, practical transactions cost structures and other real-world constraints will be formulated and used subsequently to evaluate the feasibility of the proposed multi-objective evolutionary platform that simultaneously optimized tracking performance and transaction costs throughout the investment horizon.

Conclusions are drawn in Chapter 8, where the key contributions are summarized and avenues for future works are highlighted.

1.4 Summary

In this chapter, a general introduction to investment portfolio management, particularly asset allocation and management styles, was provided. Some of the associated issues that will be investigated further in subsequent chapters were highlighted also. An overview of the thesis was provided at the end of the chapter. For brevity, many details pertaining to investment portfolio management were specifically omitted and interested readers are referred to standard textbooks for further clarification.

Chapter 2

Evolutionary Multi-Objective Optimization

2.1 Introduction

Many real-life problems require the simultaneous optimization of several non-commensurable and often competing objectives. In the context of investment portfolio management as an example, building up a portfolio that targets excess risk-adjusted yield over the aggregate market returns will encompass several objectives like maximizing the expected returns of the portfolio constituents, minimizing the aggregate returns volatility of the portfolio, minimizing the transaction costs in the purchase of the securities and etc. Besides these conflicting objectives, several constraints in accordance to the investment mandate and business/industrial regulations have to be considered, for example, maintaining specific exposure to a particular industrial sector and/or market region, limiting excessively concentrated holdings for diversification benefits and etc.

In general, multi-objective optimization (MOO) involves the balancing of the different objectives in the optimization problem, each according to their right level of importance, and search for the optimum or best compromise between them, whilst keeping within the various constraints. Comparatively, single-objective (SO) optimization is concerned with finding the one solution that optimizes the sole objective function of the underlying problem. Unlike SO optimization where a complete ordering of the solutions exists, MOO presents a possibly uncountable set of solutions that represent alternative trade-offs between the various objectives. The search for the optimal set of solutions in MOO is often an extremely difficult search problem. In fact, multi-objective problems, including the Markowitz's mean-variance model, are in general NP-complete [10].

Evolutionary optimizers, a class of stochastic search techniques, have been gaining significant attention from the research community in the field of MOO, due to its success in solving complex real-world optimization problems with various competing specifications. In fact, conventional evolutionary optimizers, including evolutionary algorithms, particle swarm optimization and ant colony optimization, and they have been extensively applied to portfolio optimization. As most of these meta-heuristics models adopt a population-based search approach, they are especially well-suited for MOO due to their ability to sample a pool of solutions simultaneously during the optimization progress.

The remainder of this chapter is organized as such. It will start with a formal definition of the key principles underlying MOO, followed by a discussion on the optimality conditions in the presence of multiple objectives. The latter part of the chapter will present a short overview on the various type of evolutionary optimizers considered in this thesis, namely evolutionary algorithms and particle swarm optimization and a brief discussion on how they can be extended for the purpose of MOO in general.

2.2 Multi-Objective Optimization

2.2.1 Problem Definition

Without any loss in generality, a minimization multi-objective problem (MOP) can be formally defined as follows [243]:

$$\min_{\vec{x} \in \Re^n} \vec{f}(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), ..., f_m(\vec{x})\} \tag{2.1}$$

$$\text{s.t. } \vec{g}(\vec{x}) > 0, \vec{h}(\vec{x}) = 0$$

where $\vec{x}$ is the vector of decision variables bounded by the decision space, $\Omega : \vec{x} \in \Re^n$ and $\vec{f}$ is the set of objectives to be minimized. The functions $\vec{g}$ and $\vec{h}$ represent the set of inequality and equality constraints that defines the feasible region of the n-dimensional continuous or discrete feasible solution space. The MOP's evaluation function,$F : \Omega \rightarrow \Lambda$, maps decision variables $\vec{x}$ to objective vectors $\vec{f}$ as illustrated in Figure 2.1 for the case where $n = 3$ and $m = 2$. Depending on the underlying objective functions and constraints of the particular MOP, this mapping might not be unique and may be one-to-many or many-to-one. The objective vectors will directly determine the optimality of the solution.

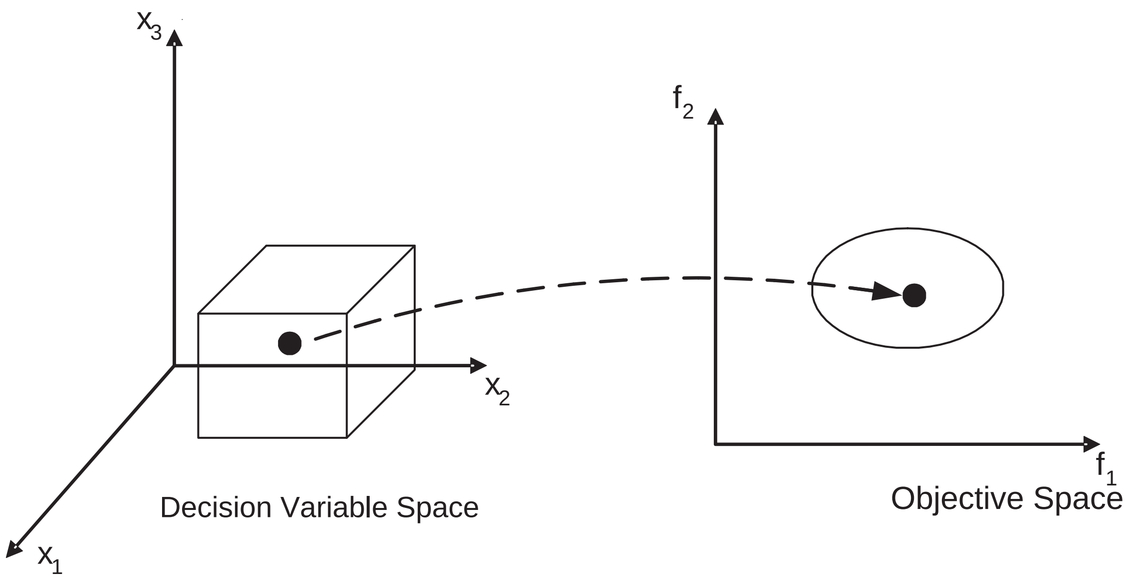


Figure 2.1: Evaluation mapping function between the decision variable space and objective space in MOO.

2.2.2 Pareto Optimality

In single objective optimization, the feasible set is completely ordered according to the objective function. For any given set of solution, the best solution can be clearly defined according to their corresponding objective value. The goal thus is to simply find the solution that maximizes/minimizes the objective function. However, in the case of MOO where several objectives are involved, the ordering of the solution set becomes complicated. Early approaches aggregated the various objectives into a single, parameterized objective function, hence congregating the multiple objectives into a single objective. Classical representatives of this class of techniques are the weighting method, the constraint method, goal programming and the min-max approach. However, limitations pertaining to these methods include high computational cost, prior knowledge of the problem required, bias towards certain regions of the trade-off curve and etc. Hence, it is imperative that an alternative notion of optimality is needed in MOO.

The Pareto optimality is a standard of judgment in which the optimum allocation of the resources are not attained as long as it is possible to make at least one individual better off in its own estimate while keeping the others as well off in their own estimate. In the realm of MOO, especially during the total absence of information regarding the preferences or importance of each objective, ranking scheme based upon the Pareto dominance is regarded as an appropriate approach to represent the strength of each individual in MOO [224]. The formal definitions of Pareto dominance are as follows [243]:

Definition 2.1: Weak Dominance:$\vec{f_1} \in \vec{F}^M$ weakly dominates $\vec{f_2} \in \vec{F}^M$, denoted by $\vec{f_1} \preceq \vec{f_2}$ iff $f_{1,i} \leq f_{2,i}\ \forall i \in \{1, 2, ..., M\}$ and $f_{1,j} < f_{2,j}\ \exists j \in \{1, 2, ..., M\}$

Definition 2.2: Strong Dominance: $\vec{f_1} \in \vec{F}^M$ strongly dominates $\vec{f_2} \in \vec{F}^M$, denoted by $\vec{f_1} \prec \vec{f_2}$ iff $f_{1,i} < f_{2,i}\ \forall i \in \{1, 2, ..., M\}$

Definition 2.3: Incomparable: $\vec{f_1} \in \vec{F}^M$ is incomparable with $\vec{f_2} \in \vec{F}^M$, denoted by $\vec{f_1} \sim \vec{f_2}$ iff $f_{1,i} > f_{2,i}\ \exists i \in \{1, 2, ..., M\}$ and $f_{1,j} < x_{2,j}\ \exists j \in \{1, 2, ..., M\}$

The various Pareto Dominance relationships are illustrated in Figure 2.2, which depicts a reference solution and four different regions highlighted in different shades of grey. Solutions located in region A dominate the reference solution as the latter is worse in both objectives when compared to the former. Similarly, solutions located in region D are dominated by the reference solution. Solutions located in regions B and C are incomparable to the reference solution because it is not possible to establish any superiority of one solution over the other i.e. solutions in the region C are better only in f_2 while solutions in the region B are better only in f_1. Lastly, solutions located at the boundaries between region B/C and D are weakly dominated by the reference solution. It can be easily noted that there is a natural ordering of these relations: $\vec{f_1} \prec \vec{f_1} \;\Rightarrow\; \vec{f_1} \preceq \vec{f_1} \;\Rightarrow\; \vec{f_1} \sim \vec{f_2}$.

With the concepts of Pareto dominance properly defined, the concept of Pareto optimality is defined as follows [243]:

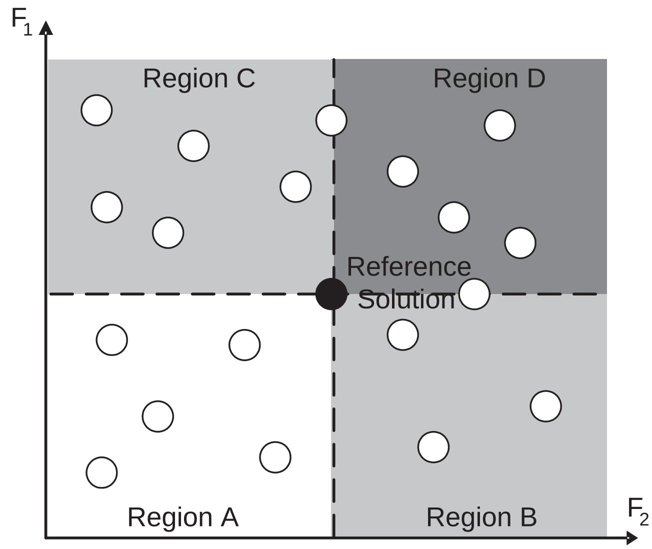


Figure 2.2: Illustration of the Pareto Dominance relationship between candidate solutions and the reference solution.

Definition 2.4: Pareto Optimal Front: The Pareto optimal front, denoted as PF_{True}, is the set of non-dominated solutions with respect to the objective space such that $\mathrm{PF}_{True} = \{\vec{f}_i^* | \nexists \vec{f}_j \prec \vec{f}_i^*, \vec{f}_j \in \vec{F}^m\}$

Definition 2.5: Pareto Optimal Set: The Pareto optimal set, denoted as P_{True}, is the set of solutions that are non-dominated in the objective space such that $\mathrm{P}_{True} = \{\vec{x}_i^* | \nexists \vec{F}(\vec{x}_j) \prec \vec{F}(\vec{x}_i^*), \vec{F}(\vec{x}_j) \in \vec{F}^m\}$

The various concepts of Pareto optimality are illustrated in Figure 2.3. Solutions lying on the boundary between the infeasible region and feasible region are Pareto Optimal with respect to decision search space, since no other solutions can possibly dominate them. This boundary represents PF_{True} that corresponds to the set of solutions in P_{True}.

2.2.3 Optimization Goals

Clearly, it will be impossible to find the entire P_{True} which most likely constitutes infinite elements. On a more practical note, what can be done instead is to find a set of solutions, P_{Known} within the limited computational resources, which when plotted in the objective space, generates a Pareto front, PF_{Known} that can best represent PF_{True}. An example of such an approximation is illustrated by the

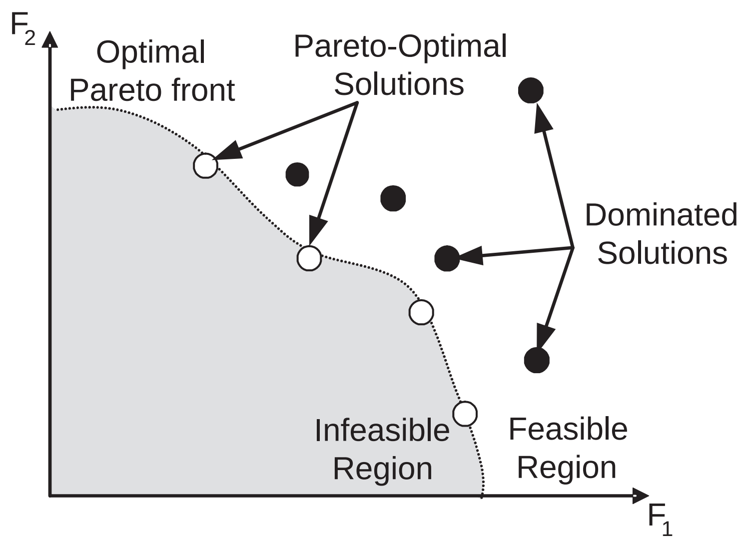


Figure 2.3: Illustration of the various concepts of Pareto Optimality.

set of non-dominated solutions denoted by the circles residing along the PF_{True} in Figure 2.3. The definition of quality of the discovered solution set, PF_{Known} contains multiple criteria [43, 62, 261]:

- Proximity: Minimizing the distance between PF_{True} and PF_{Known}.
- Spread: Maximizing the spread of the solution in PF_{Known}.
- Spacing: Obtaining a good distribution of generated solutions along PF_{Known}.

Figure 2.4 compares two different sets of PF_{Known} and the plots illustrate the superiority of one set over the other in each of the optimization goals. While the first optimization goal of convergence is the first and foremost consideration for all optimization problems in general, the second and third optimization goals of maximizing diversity are entirely unique to MOO. The rationale of finding a diverse and uniformly distributed PF_{Known} is to provide the decision maker sufficient information about the trade-offs between the different solutions before the final decision is made. It should also be noted that the optimization goals of convergence and diversity are inherently conflicting in nature, which explains why MOO is much more difficult than single-objective optimization.

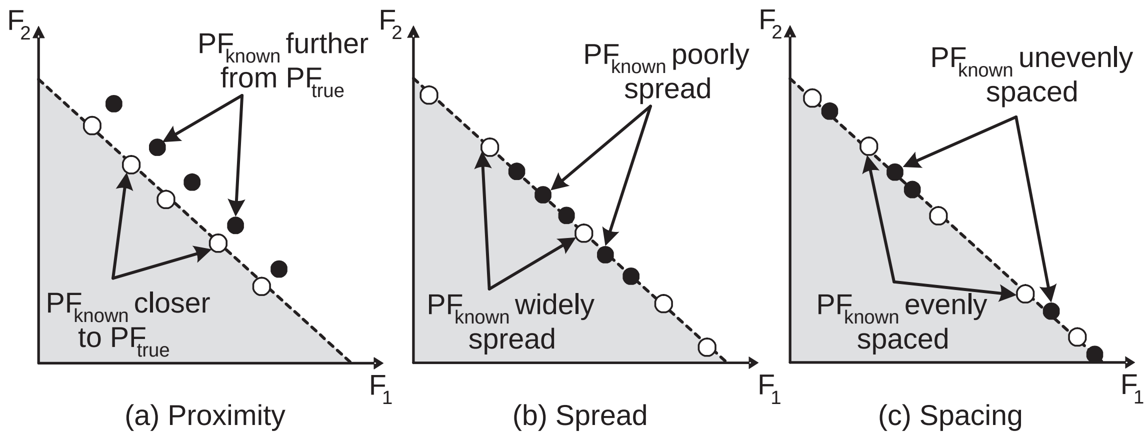


Figure 2.4: Plots comparing two different sets of solutions (white circles versus black circles), where each plot illustrates the superiority of the set of white circles over the black circles in terms of (a) proximity, (b) Spread and (c) Spacing.

2.3 Evolutionary Optimization

Traditional operational research approaches in MOO typically entails the transformation of the original problem into a single-objective problem and employs point-by-point algorithms such as branch-and-bound to iteratively obtain a better solution. Such approaches have several limitations including the requirement of the multi-objective problems to be well-behaved (i.e. differentiability or satisfying the Kuhn-Tucker conditions), sensitivity to the shape of the PF_{True} and the generation of only one solution for each simulation run. On the other hand, evolutionary optimizers that are inspired by biological or physical phenomena have been gaining increasing acceptance as a flexible and effective alternative to such optimization problems in the recent years.

2.3.1 Evolutionary Algorithm

Evolutionary algorithm (EA) stands for a class of stochastic optimization methods that adopts Darwin's principle on "survival of the fittest" and emulate the natural biological evolution mechanism. Technically, EA comprises of several evolutionary meta-heuristics model, namely, genetic algorithm [108], evolutionary programming [84] and evolutionary strategy [196]. Interestingly, evolutionary strategy is the only paradigm developed for the purpose of optimization; genetic algorithm was designed as a general adaptive system while evolutionary programming is developed as a learning process to create artificial intelligence. However, no distinction will be made between these different evolutionary computation models and they will be collectively known as EA here.

Using strong simplifications, this approach modifies a set of candidate solutions based on the two basic principles of evolution: selection and variation. Selection represents the competition for resources among living beings in which the better ones are more likely to survive and pass down their genetic information. This is simulated via a stochastic selection process, where each solution is given a chance to reproduce a certain number of times, dependent on their quality. The other principle, variation, imitates natural capability of creating "new" living beings by means of recombination and mutation. In this context, it is concerned with how potential new solutions can be generated from existing solutions at hand.

Essentially, EA maintains a population of individuals and each individual represents a possible solution to the optimization problem at hand. These individuals are encoded as chromosomes to epitome the mechanics of DNA blueprint for living organisms, allowing the propagation of information through variation and the inheritance of desirable properties by offspring solutions. When decoded, they generate a set of decision variables which represent a particular point in the objective function space. The optimality of each chromosome can thus be determined, depending on how "well" the various constraints and objectives in the problem are satisfied.

The algorithmic flow of the general EA is illustrated in Figure 2.5. It will start by initializing a random population of candidate solutions. Based on their fitness, the better individuals will be selected as parents to seed the next generation. Variation operation will subsequently be applied to them to generate a new set of candidate solutions. These offspring will compete with the old individuals based on their fitness for a place in the next generation. By subjecting the population of individuals through this process for generations, the individuals will evolve to adapt to the environment, accompanied by an overall rise in the fitness level of the population. This cycle will terminate when either a set of candidate solutions with sufficient quality had been found or a predefined computational limit had been reached. The archive represents the elitist strategy [60], which is used to preserve the best individuals found into the next generation. The underlying motivation is to prevent the lost of good individuals due to the stochastic nature of the evolution process. De Jong

[60] found that elitism could improve the performance of EAs in general although there is a potential danger of premature convergence.

```
P       Initialization (a random population )
A       Create Archive
While   (Stopping criteria not satisfied)
            P    Evaluate (P)
            A    Update(P, A)
            S    Selection(P, A)
            P    Variation(S)
End While
```

Figure 2.5: Algorithmic flow of a general MOEA presented as a flowchart.

Selection and variation represent the underlying force driving the search dynamics of EA. The former removed low-quality individuals from the population, so that high quality individuals have a higher chance to be reproduced. This has the effect to focus the search on particular portions of the search space and to increase the average quality within the population. Mimicking the stochastic nature of evolution, the latter generates new solutions within the search space by the variation of existing ones. While selection acts as a force pushing for quality, variation creates novelty [73]. Their combined effect generally leads to improved fitness values during runtime.

Although the underlying principles are simple, these algorithms have proven themselves as a general, robust and powerful search mechanism. The strength of EA lies in their population-based search approach, which will generate higher diversity in the search space and reduce the likelihood to converge to the local optimum. However, this easily translates to higher computational costs for administrating the population pool.

2.3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a form of population (swarm)-based optimization technique developed by Kennedy and Eberhart [128], which is inspired by the social behaviors of animals. In PSO, the position of the particles denotes candidate solutions to the optimization problem and their movements, influenced by its current position, memory and social knowledge of the swarm, are regarded as the search process for better solutions. PSO operates based on the social adaptation of information, where the memory of each particle allows knowledge of good solutions, i.e. its previous best position and the location of the global best solution, to be retained throughout the entire optimization process. This allows constructive cooperation between particles, as particles in the swarm share information between them [121].

Standard particle swarm optimizer maintains a swarm of particles that represent the potential solutions to the problem on hand. Each particle $\vec{x} = \{x_1, x_2, ..., x_n\}$ embeds the relevant information regarding the decision variables and is associated with a fitness that provides an indication of its performance in the objective space. Each particle will keep track of its previous best position (*pbest*), denoted by $\vec{p}_b = \{p_{b,1}, p_{b,2}, ..., p_{b,n}\}$ and its corresponding fitness. Apart from *pbest*, each particle also has knowledge on the best position found so far by all the solutions (*gbest*), denoted by $\vec{p}_g = \{p_{g,1}, p_{g,2}, ..., p_{g,n}\}$.

In essence, the trajectory of each particle is updated according to its own flying experience as well as to that of the best particle in the swarm. At each time step, t, each particle will be accelerated towards its *pbest* and the *gbest*. The velocity update equation (2.2) and position update equation (2.3) are described as follows:

$$\vec{v}(t) = I \times \vec{v}(t-1) + c_1 \times rand() \times (\vec{p}_b - \vec{x}(t-1)) + c_2 \times rand() \times (\vec{p}_g - \vec{x}(t-1)) \tag{2.2}$$

$$\vec{x}(t) = \vec{x}(t-1) + \vec{v}(t) \tag{2.3}$$

where I is the inertia weight which balances the global exploitation and local exploration abilities of the particles, c_l and c_2 are acceleration constants, $rand()$ are random values between 0 and 1. Iterative updating of their positions based on (2.2) and (2.3) will result in the entire swarm "flocking" towards the optimal vector, whilst each particle moving randomly.

The major strength of PSO lies in their simplicity in implementation and high computational efficiency in solving optimization problems [129]. The nature of their representation makes them well suited for numerical optimization problems, which in the context of portfolio optimization, will be applicable in optimizing the asset allocation aspect of the problem.

2.3.3 Multi-Objective Evolutionary Algorithm

Since the pioneering effort of Schaffer [209], many different evolutionary techniques for MOO have been proposed with the aim of fulfilling the three optimization goals described previously. While most of the early works are largely based on the computational models of genetic algorithm [108], evolutionary programming [84] and evolutionary strategy [196], other biologically inspired models such as particle swarm optimization , differential evolution , cultural algorithms , and artificial immune systems have been extended for MOO in recent years. Though these algorithms are distinctively different in methodology, their distinctions between them have become increasingly vague as researchers sought to exploit the advantages offered by the different algorithms in a common platform. Issues on hybridizing different evolutionary paradigm, which is otherwise known as Memetic algorithms will be discussed in greater details in the next subsection.

As highlighted earlier, MOO requires researchers to address many similar issues that are unique to multi-objective problems for example, maintaining the diversity of the PF_{known}. These issues are invariant to the type of evolutionary computation model applied. Therefore, no distinction will be made between them and these techniques developed for MO optimization are collectively referred to as multi-objective evolutionary algorithm (MOEA) in this thesis.

The general MOEA framework is identical to the pseudo code shown in Figure 2.5. EA and MOEA are essentially similar with both models involving an iterative adaptation of a set of solutions until a pre-specified optimization goal/stopping criterion is met. What sets these techniques apart is the increased emphasis on diversity in the solution set by the latter. This is actually a consequence of the optimization goals described in the previous section. Particularly, the search dynamics must drive the solutions toward the PF_{true} as well as distribute the individuals uniformly along the discovered PF_{known}. The evolutionary operators and elitism strategy updating must also take diversity into consideration to encourage and maintain a diverse solution set.

According to [159], simple MOEA tends to converge towards a single solution, failing the MOO goals in achieving a good spread and distribution in the obtained PFknown. This necessitated the development of diversity operators that can maintain substantial amount of diversity in the evolving population, allowing the MOEA to perform a multi-directional search simultaneously to discover multiple, widely different solutions. Depending on the manner in which solution density is measured, the different density assessment techniques can be broadly categorized under distance-based, grid-based, and distribution-based. Interested readers are referred to [60, 100, 131] which introduce some of the most frequently used diversity operators in evolutionary MOO.

One of the main design issues of diversity operators is whether the density assessment is computed in the decision space or objective space. While some works advocated the former since this is where the decision variables are searched upon [224], most work reported in the literature applied density assessment in the objective space due to the optimization goals of obtaining a well-distributed PFknown. There has been works that considered density assessment in both objective and decision spaces simultaneously [9, 107, 203]. Nevertheless, Tan et al [232] pointed out that the choice essentially depends on what is desired for the underlying problem.

Elitist strategy represents an important component of MOEA due to its significant impact on the algorithmic performance [261]. In fact, various studies have shown that elitism is a theoretical necessity for MOEA convergence [134, 204, 205]. Elitism involves two closely related processes, namely, the preservation of good solutions and the reinsertion of these solutions into the evolving population.

The first issue is concerned with the type of solutions to be stored in the archive. This issue is further complicated by the restriction in archive size due to limited computing and memory resources in practical implementation. Most works enforce a bounded set of elitist solutions which requires a truncation process when the size of the elitist solutions exceeds a predetermined bound. This process is normally based on some form of density assessment discussed earlier. However, other measures such as hypervolume [133] and relaxed forms of Pareto dominance have been applied as well [64, 190].

The second issue is concerned with the "exploration versus exploitation" dilemma, where a higher degree of exploitation attained through the reintroduction of elitist solutions leads to the lost of diversity while too much exploration leads to slow convergence rates. Insufficient diversity will result in MOEA failing to attain a well-distributed PF_{Known}, and in the worst case, premature convergence to local optimal solutions

2.3.4 Memetic Algorithm

The design of global optimization techniques is governed by two competing goals, namely global reliability and local refinement [239]. The former is required to ensure that every region of the search space is covered to provide a reliable estimate of the global optimum and the latter to further refine the good solutions by concentrating search effort around their neighborhood. Many global optimization techniques achieved these two goals by adopting a combination of global and local search strategy.

Although evolutionary meta-heuristics have shown general success in solving complex real-world optimization problems with various conflicting specifications, they suffer from slow convergence towards the optimum due to their strong stochastic bias [185, 95] and poor accuracy of the evolved solutions due to the lack of local refinement [141]. Memetic algorithms (MA), motivated by the apparent need to employ both global and local search strategy to provide an effective global optimization method, extend evolutionary meta-heuristics by incorporating local search operators to complement the evolutionary operators. With this synergetic combination, the evolutionary operators will perform an adaptive, global sampling of the search space that actively generates solutions in new basins of attraction [104] throughout the evolution, while the local optimizers will efficiently refine these solutions and identify the corresponding local optimum within each basin. Studies have shown the relative advantages of MA over evolutionary meta-heuristics in terms of solution quality and computational efficiency [14, 176, 184, 185, 187, 259, 260].

Ong and Keane [185] investigated the suitability effects of the local search operators (along with the appropriate parameter setting) for the optimization problem in hand and, backed by empirical studies [104, 140], commented that without the correct application of local search, MA may not perform at its optimal level or worse, it may perform poorer than using the evolutionary meta-heuristics alone. Hart [104] brought up several questions that should be addressed in designing local optimizers for evolutionary search. Chiam et al. [46] extend these questions and proposes four main issues involving the implementation of evolutionary local search, namely as follows:

- Type of local optimizer
- Integration approach
- Candidate solutions for local search
- Computational balance between global and local search

The varied suitability of different local search strategy for various optimization problems has consequently motivated the development of a wide variety of local optimizer in literature. Early local optimizers were just simple random perturbation about the search space by adding zero mean normal deviates to the different dimensions of the candidate solution. Subsequently, motivated by the steepest descent method, local optimizers used gradient information and perturbed solutions in preferred direction towards the optimum. However, the former might introduce inefficiency into the algorithm, while the latter will fail in discontinuous or non-differentiable search space. Of more recent issue, evolutionary optimizers like PSO, ant colony optimization [102, 149, 258] and even EA itself [68, 214, 257] are increasingly being used directly as local optimizer itself, though this will generally increase computational cost [56] due to higher computational overhead.

The next design issue addresses how the local optimizer is being integrated into the evolutionary search. Some of the possible approaches include two-phase implementation, Lamarckian learning and Baldwin learning. In the first approach, the evolutionary search will be employed first to identify promising regions containing the global optimum, and then local search will be applied for fine tuning [98, 193, 199]. Studies have been conducted to validate the two-phase approach and the simulation results have been promising [14, 132, 220]. However, Ku et al. [141] highlighted that most of the optimization problems used in these studies could be handled easily by the local search operator itself. As such, the true capability of two-phase implementation can only be evaluated with test problems that cannot be solved by the local search strategy alone.

Lamarckian learning is based on the heritability of acquired characteristics, where an organism can pass on characteristics that it acquired during its lifetime to its offspring. The application of Lamarckian learning in evolutionary search is rather straightforward. Local search will be applied to candidate solutions at pre-specified intervals of the evolutionary search and if their fitness improved, these solutions (together with their associated fitness) will be returned to the population for further evolution [25, 255] . In contrast to Lamarckian learning, only the improved fitness is inherited by the children after the local search in Baldwinian learning, hence the learned traits are lost. While the capability of Baldwinian learning has been demonstrated in several instances [1, 106], some empirical studies have demonstrated its inefficiency in wasting precious computational resource [85, 125, 248].

The third issues tackle the selection issue of choosing appropriate solutions from the evolving population for local enhancement [104]. The candidate solutions for the local search could be either the best solution obtained by the evolutionary search [173], offspring solutions that satisfy certain requirements or simply every child generated by the evolutionary operators. Of course, every scheme has their strengths and limitations. For example, the infrequent activation of local search in the first scheme might not provide significant improvement [173], while the third scheme might reduce the overall computational efficiency [101] by wasting precious computational resources on solutions with low likelihood of becoming best solution. Lastly, for the second scheme, an appropriate criteria to determine suitable candidate solutions is required, otherwise benefits gained from local optimization might not be significant [138].

Although the application of local search operators will usually improve algorithmic performance, it might increase computational cost in terms of higher fitness evaluation, depending on their efficacy and efficiency. As such, it is important to determine how the limited computation resource should be allocated between the evolutionary search and local search i.e. the frequency in which the local search is triggered and the number of fitness evaluations to be included in each iteration [104]. A delicate balance should be maintained between evolutionary search and local search to prevent any under-utilization in either of the search strategy [113] . This is especially important for population-based optimizer like EA and PSO due to their high computational overhead. The manner to adjust the computational balance depends on how local search is invoked during the evolutionary search, for example the local search probability [113, 189] , the local/global ratio [144] , the size and type of neighborhood structure [140] and etc.

2.4 Summary

Most real-world optimization problems naturally comprised of multiple objectives that are inherently conflicting to each other. A formal definition of multi-objective optimization was presented in this chapter, together with concepts of Pareto dominance, which is used to quantify the optimality of solutions in the presence of multiple objectives. Under time and resources limitations, related works in MOO have increasingly turned to evolutionary computation, which has proven to be an effective and efficient optimization tool for this purpose. A brief overview on the different variants of evolutionary optimizers that will be considered in this work was also presented.

Chapter 3

Extending MOEA for Portfolio Optimization

3.1 Introduction

This chapter will look at how the generic MOEA platform can be extended for the purpose of portfolio optimization. The central theme in portfolio optimization involves asset selection and allocation i.e. selecting the appropriate subset to be included in the portfolio and determining their corresponding proportions. The first two sections will propose an order-based chromosomal representation for portfolio and a set of variations operator specific to the proposed chromosomal data structure. The later half of this chapter will focus instead on algorithmic enhancements to the MOEA platform. Specifically, an PSO-EA memetic model will be introduced that can improve overall algorithmic performance via a local search strategy of fine-tuning asset weights of the portfolio. Next, dynamic optimization will be considered and the proposed dynamic archiving will be useful in later chapters where multi-period portfolio optimization is examined.

3.2 Chromosomal Representation for Portfolio Structure

MOEA maintains a population of chromosome, where each of them represents a potential solution to the optimization problem, which in the context of portfolio management is a portfolio of assets.

In related literature, different types of representation have been proposed. The most direct representation is to use a real-number vector that denotes the weight composition of the various assets in the portfolio [65]. Before the fitness evaluation, the total weight is normalized to one to satisfy the budget constraint.

However, better algorithmic performance can be obtained, if a problem specific representation is adopted instead. Streichert et. al [225] observed that the optimal portfolio normally comprised of only a limited number of the available assets. As such, a hybrid representation was proposed, where an additional binary string is included to reflect the existence of the assets in the portfolio. Such a scheme facilitates the removal and adding of assets to portfolios, resulting in smaller portfolios generally. This representation has been popular in related literature [221, 227, 225, 240]. Alternatively, the weight vector can just comprise of a few assets that are randomly chosen prior to the algorithmic run [7, 174] . This approach provides a simple solution to the fixed cardinality constraint that limits the portfolio size to a particular value.

3.2.1 Order-Based Representation

An alternative representation for portfolio management will be introduced here. It is essentially an order-based representation, where each chromosome comprises of two vectors, an integer vector that contains the identity tags of the various assets available and a real number vector, denoting their corresponding weights. Figure 3.1 shows an instance of this representation for a problem with eight assets available.

Asset Vector	7	2	5	1	3	8	6	4
Weight Vector	0.54	0.25	0.85	0.14	0.05	0.67	0.55	0.40

Figure 3.1: A chromosomal instance for the ordered based representation proposed based on eight assets available.

To find the portfolio associated with this chromosome, an empty portfolio will first be initialized and assets be added iteratively, as per their order in the asset vector. This procedure will terminate once the accumulated weight of the portfolio exceed one or when all the available assets are in the portfolio. The total weights for the assets in the portfolio will then be normalized to one to satisfy the budget constraint. After which, the associated fitness functions can be evaluated to determine its optimality. The number of assets added to the empty portfolio denotes the portfolio size or cardinal. Figure 3.2 illustrates the fitness evaluation procedures for the chromosome in Figure 3.1, where the assets are added to the empty portfolio until the third assets as the accumulated weights exceed one.

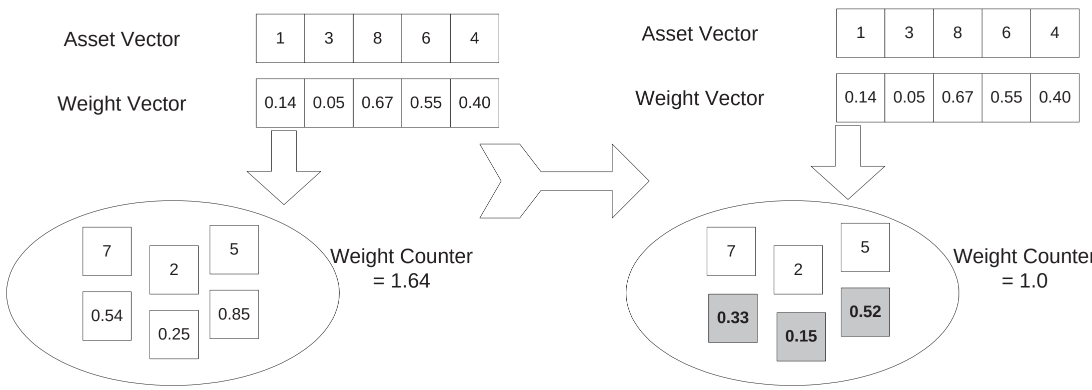


Figure 3.2: Fitness evaluation for the chromosome in Figure 3.1. Assets are iteratively added into the portfolio until the accumulated weights exceed one. The various weights in the portfolio are then normalized to one to satisfy the budget constraint.

As each asset is added iteratively into the portfolio for the order-based representation, direct monitoring and control of the weight values for each asset is possible at their point of inclusion. As such, constraint handling techniques can be executed instantaneously to repair any infeasibility. Further details of the constraint handling techniques will be furnished in the later chapters.

3.2.2 Empirical Study & Analysis

The chromosome initialization process involves randomly permuting the order of the asset vector and generating the weights from certain probability distribution. The most direct approach is to adopt the uniform distribution ranging from 0 to 1.

$$w_i = U(0,1) \tag{3.1}$$

where w_i denotes the weight value of the $i-th$ asset and $U(0,1)$ represents the uniform distribution function on the interval $[0,1]$. However, this will correspond to a mean weight of 0.5, which implies that on average, only two to three assets are required to fill up an empty portfolio. As such, simple implementation of this representation will tend to generate portfolio of smaller cardinality. Specifically, the average portfolio size for 100,000 randomly generated chromosomes is around 2.7 with a standard deviation of 0.9.

A simple solution to increase the average portfolio size will be to impose a maximum limit for the various weight values during initialization, as follows:

$$w_i = U(0,1) \times W_{Max} \tag{3.2}$$

where W_{Max} denotes the maximum weight value. The average size of the portfolio generated will vary with the value of W_{Max}. For example, the average portfolio size for 100,000 randomly generated chromosomes with W_{Max} of 0.1 is around 20.7 with a standard deviation of 2.6. To investigate further, a set of values were considered for $W_{Max} \in \{5.0, 2.0, 1.0, 0.5, 0.2, 0.1, 0.05\}$ and 100,000 chromosomes of maximum length 30 were randomly generated. Figure 3.3 illustrates the boxplot depicting the distribution of the portfolio size for the chromosomes under different values of W_{Max}. Evidently, lower W_{Max} will translate to larger portfolio sizes. Even though, the weights are capped at a limit of 1.0, larger W_{Max} were considered also for investigative purposes and in such cases, the average portfolio sizes actually decreases. Nevertheless, although the imposition of W_{Max} can help to tune the portfolio cardinality, the diversity of the initial population remained fairly low. "$0 \sim 0.2$ " in Figure 3.3 represents the case where each chromosome is assigned a different W_{Max} value, derived from a uniform distribution on the interval $[0, 0.2]$. Interestingly, a random W_{Max} will arbitrarily enhance the diversity of the initial population.

To put the earlier results into perspective, it will be instructive to compare the distribution of portfolio size within a random population generated via the different representation schemes. Specifically, real-number representation, hybrid representation and several variants of order-based representation with different W_{Max} were considered. Table 3.1 summarizes the statistical information on the portfolio size distribution under the various representation schemes. Expectedly, real-number representation resulted in portfolio comprising of all the different assets due to the low probability of generating a zero to completely exclude the asset from the portfolio. Conversely, hybrid representation has an average portfolio size of 15, due to the 50% probability of obtaining a '1 'in the

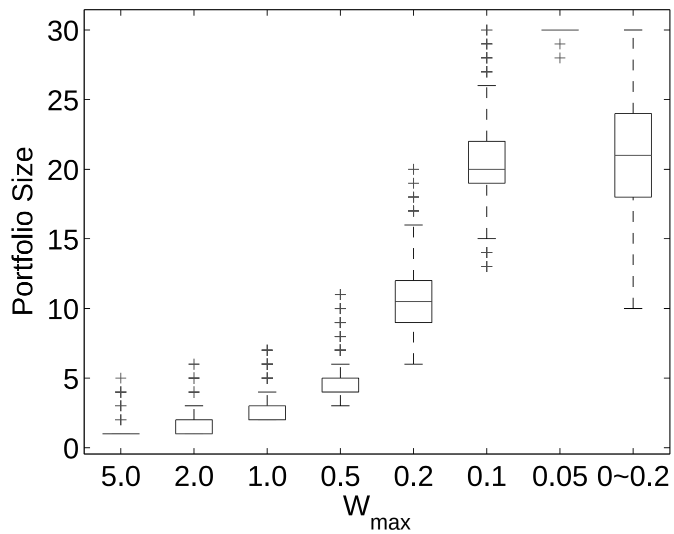


Figure 3.3: Average portfolio size (maximum 30) for 100,000 randomly generated chromosomes with different weight limits. '$0-0.2$'denotes the case where each chromosome is assigned a different Wmax value, derived from a uniform distribution on the interval [0, 0.2].

binary vector. Adopting a random W_{Max} will improve the standard deviation of the cardinality distribution. Figure 3.4 provides a graphical illustration on the frequency distribution of the portfolio size for the different representation schemes. Clearly, the application of a random W_{Max} results in better population distribution in terms of diversity and spread.

At times, it will be useful to generate portfolios with specific size, for example in addressing cardinality constraints or directing the evolutionary search towards particular region in the search space. The empirical results earlier revealed that the average portfolio size of random chromosome can be tuned by varying W_{Max}. While W_{Max} of 1 will result in an average portfolio size of 2 to 3, decreasing W_{Max} to 0.1 will increase average portfolio size to around 20 to 21. However, from a user's perspective, it will be more convenient to specify a target portfolio size (K_{Target}), instead of tuning W_{Max}.

Theoretically, to obtain an average portfolio size of K_{Target}, the expected value of each weight should be $1/\ K_{Target}$. Generally, this can be done by considering any random distribution with mean $1/\ K_{Target}$. One instance of such function can be obtained by a simple modification of (3.2),

Table 3.1: Statistical Information on the portfolio size (maximum 30) for 100,000 randomly generated chromosomes with different representation schemes.

Representation	Portfolio Size (Average)	Portfolio Size (Standard Deviation)
Real Number Representation (RR)	30.0	0.0
Hybrid Representation (HR)	15.0431	2.7467
Order-based representation with W_{Max} of 1.0 (ORW_1)	2.7209	0.8769
Order-based representation with W_{Max} of 0.5 ($\text{ORW}_{0.5}$)	4.6695	1.2415
Order-based representation with W_{Max} of 0.1 ($\text{ORW}_{0.1}$)	20.6711	2.6236
Order-based representation with random W_{Max} between 0 and 0.2 ($\text{ORW}_{0\sim 0.2}$)	20.8934	3.9084

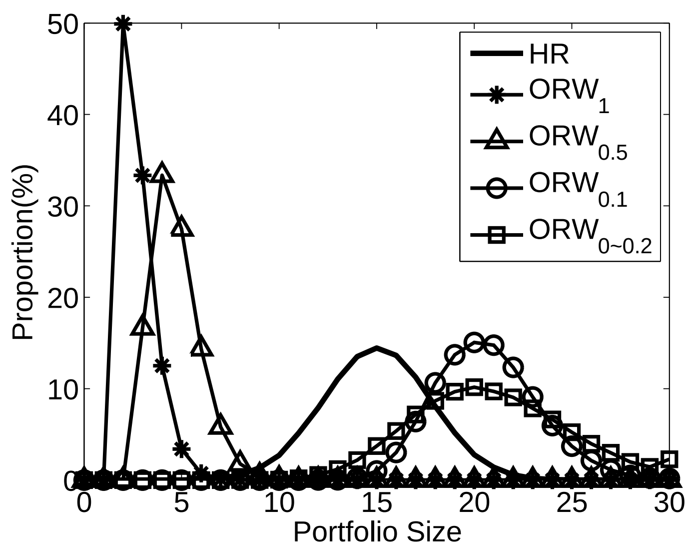


Figure 3.4: Distribution of portfolio size (maximum 30) for 100,000 randomly generated chromosomes with different representation schemes.

by defining an appropriate value for W_{Max} as shown below

$$w_i = U(0,1) \times \frac{2}{K_{Target}} \tag{3.3}$$

In this case, the expected value of each weight will just be, $E(w_i) = 0.5 \times \frac{2}{K_{Target}} = \frac{1}{K_{Target}}$, resulting in the targeted portfolio size of K_{Target}. The simulation setup earlier will be repeated here to evaluate the validity of (3.3). Considering various values of K_{Target} ranging from 2 to 30 in step size of 2, the corresponding population distribution of the portfolio size were plotted in Figure 3.5. Clearly, the average portfolio size corresponds accurately to the pre-specified K_{Target}, exhibiting the flexibility and extensibility of OR in practical implementation. It might be expedient at times to

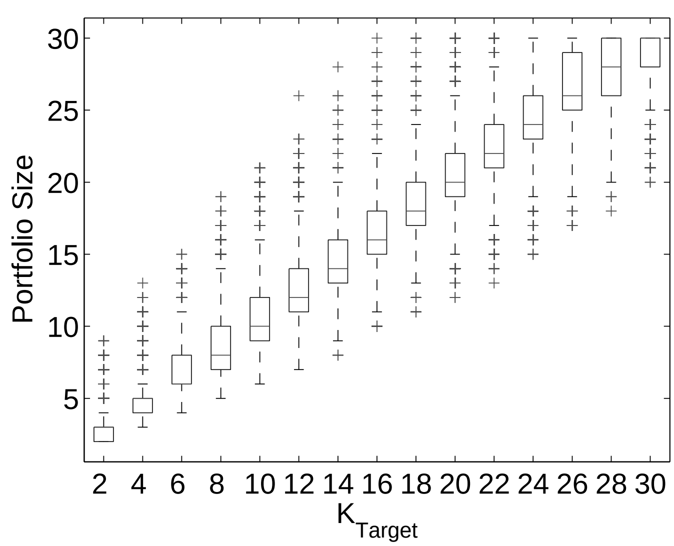


Figure 3.5: Box plots illustrating the portfolio size distribution (maximum 30) for 100,000 randomly generated chromosomes with different value of K_{Target}.

define a targeted range instead of a specific value. For this purpose, the following weight generation function is proposed

$$w_i = U(0,1) \times \frac{2}{U(K_{MinTarget}, K_{MaxTarget})} \tag{3.4}$$

where $K_{MaxTarget}$ and $K_{MinTarget}$ respectively denotes the maximum and minimum K_{Target}. To evaluate the validity of (3.4), three different target ranges i.e. {10, 20}, {5, 10} and {20, 25} were considered (respectively denoted by ORK_{10-20}, ORK_{5-10} and ORK_{20-25}) and the corresponding frequency distribution were plotted in Figure 3.6. K_{Target} of 15 (ORK_{15}) was included also for comparison purposes. Expectedly, (3.4) allows the initialization of a portfolio set with cardinal falling within the pre-specified targeted range. This further exhibit the flexibility and extensibility of the proposed order-based representation.

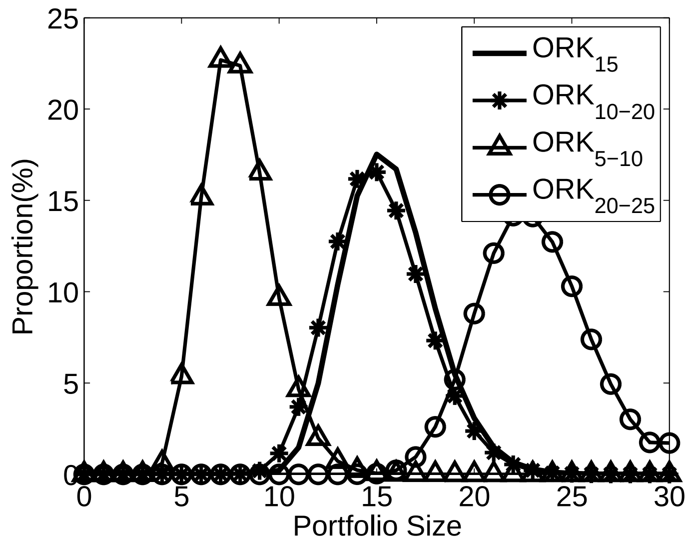


Figure 3.6: Distribution of portfolio size (maximum 30) for 100,000 randomly generated chromosomes with different targeted range.

3.3 Variation Operation

3.3.1 Crossover and Mutation Operators

Since conventional crossover or mutation operators are not suitable for the data structure of the order-based representation, representation-specific variation operators have to be designed. The proposed crossover operation is illustrated in Figure 3.7. Given two parent chromosomes, a crossover point will be randomly selected. Each chromosome will retain their original value before the crossover point and the remaining values after it will be rearranged in accordance with the order in the other chromosome, i.e. the alleles {5,2,7} in chromosome 1 are rearranged to {2,5,7}, in the order in which these three values appear in chromosome 2. The corresponding weight vector will be reshuffled accordingly also.

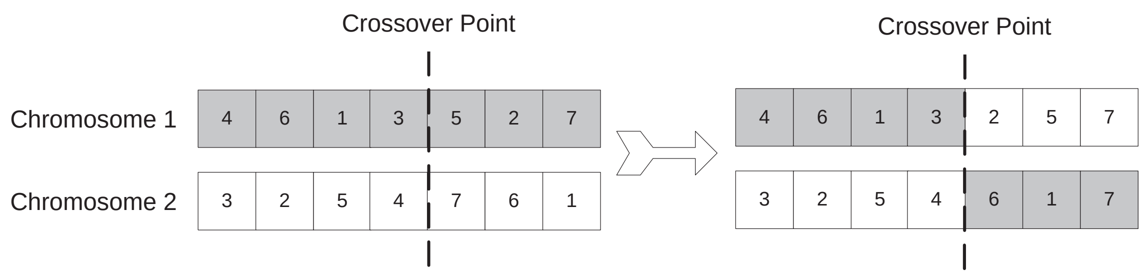


Figure 3.7: Single-point crossover. Genes after the crossover point are swapped between the two parent chromosomes.

However, not all assets in the chromosome are included in the portfolio as illustrated in Figure 3.2. As such, neutral variation [202] (where redundancy in the genotype nullified the effects of variation) might arise if the crossover point is selected amongst them. While advocates of neutral variation argued that the increased connectivity of the search space will enhance the phenotype reachability [71], hence improving the overall evolutionary search [53, 92, 124], some empirical studies argued the contrary [76, 201] . Nevertheless this topic remains highly debatable and interested readers are refereed to [45, 202] for further discussion. To isolate any algorithmic influence based on neutral variation, the crossover point is selected only amongst the assets considered in the portfolio. Specifically, the crossover point will be chosen within the mean portfolio sizes for the two parent chromosomes.

The mutation operation is just a simple procedure of swapping the asset and weights of two randomly selected alleles in a single chromosome, as illustrated in Figure 3.8. Subsequently, standard Gaussian mutation will be applied to the corresponding weights values. Again to prevent neutral variation, it should be ensured that at least one of the selected assets should be within the portfolio. Both the variation operations discussed earlier are typically used for order-based representation.

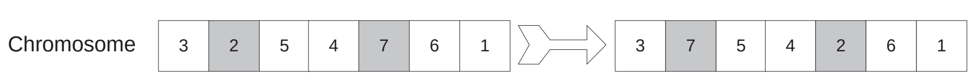


Figure 3.8: Bit-swap mutation i.e. position of randomly chosen genes are swapped.

3.3.2 Empirical Study & Analysis

To compare the algorithmic characteristic of the proposed variation operators for the order-based representation with other conventional variation operators, quantitative analysis, in the form of mutation and crossover innovation [192] , were conducted. Before discussing the simulation specifics, it is necessary to introduce metrics to quantify the distance in the phenotype search space. Distance in the phenotype search space will be measured as follows:

$$D(\vec{x}, \vec{y}) = \sum_{i=1}^{N} \mid x_i - y_i \mid \tag{3.5}$$

where $\vec{x}$ and $\vec{y}$ are two real-number vectors of length N, denoting the weights composition of the N different assets in a portfolio.

Mutation innovation (MI), which measures the phenotype distance between a pair of mutated and original solution, is a random variable that quantifies the degree of "innovation" being introduced by the mutation operator and its distribution can reflect several locality properties. Generally, higher MI will be desirable to enhance exploration efforts in the evolutionary search while lower MI will be suitable for local exploitation.

The different mutation operators compared in this simulation study are summarized in Table 3.2. Essentially, different mutation operators under the various representation schemes were considered. Mutation on binary and real values respectively involved a bit-flip operation and Gaussian mutation. P_m was arbitrarily set at 0.05.

100,000 chromosomes were randomly generated and the various mutation operators were applied. Their MI distribution were computed and plotted in Figure 3.9(a). Since RR involves only the permutation of the weight vectors, movement in the prototype search space were restricted to the tuning of the weight composition of the various assets in the portfolio. Consequently, MI for RR averaged between 0 and 1. Conversely, with the inclusion of the binary vector in HR, asset can be removed or added into the portfolio simply by inverting the binary alleles. The resultant MI distribution is more evenly spread. Generally, HR_{Same} and HR_{Diff} have similar MI distribution i.e. peak at one and converge to 0 for larger values of MI. Unlike the previous mutation operators, the MI distribution of OR_{Swap} and OR_{Mut} dipped when MI equal to 1. This is because the swap operation

Table 3.2: Description of the various mutation operators in comparison.

Notation	Representation Schemes	Description of mutation operators
RR	Real Representation	Each weight value will be mutated with a probability of P_m
HR_{Same}	Hybrid Representation	Both the binary and weight value of an allele will be mutated together with a probability of P_m
HR_{Diff}	Hybrid Representation	Each binary and weight value will be mutated independently with a probability of P_m
OR_{Swap}	Order-based Representation	Both the binary and weight value of an allele will be swapped by another randomly chosen allele with a probability of P_m
OR_{Mut}	Order-based Representation	Both the binary and weight value of an allele will be swapped by another randomly chosen allele with a probability of P_m, followed by a mutation of the weight value

requires two assets in the portfolio to be swapped, thus translating to a movement of at least 2 units in the search space. Overall, OR_{Mut}, generated larger degree of movement in the phenotype search space.

According to Raidl and Gottleib [192], it will be instructive also to investigate the case where the chromosomes are actually modified after the mutation operation i.e. MI$>$ 0. The conditional expected values E(MI|MI$>$ 0) and standard deviations σ(MI|MI$>$ 0) were calculated and listed in Table 3.3, with the corresponding distribution plotted in Figure 3.9(b). Adhering to the earlier simulation results, RR attained considerably lower E(MI) and E(MI|MI$>$ 0) while OR_{mut} attained significantly higher E(MI|MI$>$ 0), after isolating chromosomes that are not affected by the mutation operation.

As mutation is iteratively applied during the evolutionary search progress, the distribution of MI after successive mutation should be studied also. For this purpose, the phenotype distance between the original solutions and the one created after $k \in \{1, 2, 3, 4, 8, 16, 32, 64, 128, 256, 512, 1000\}$ mutations were calculated, denoted as MI^k. The empirically obtained means values, $E(MI^k|MI^k > 0)$ and standard deviations, $\sigma(MI^k|MI^k > 0)$ over the number of mutation k were plotted in Figure 3.10.

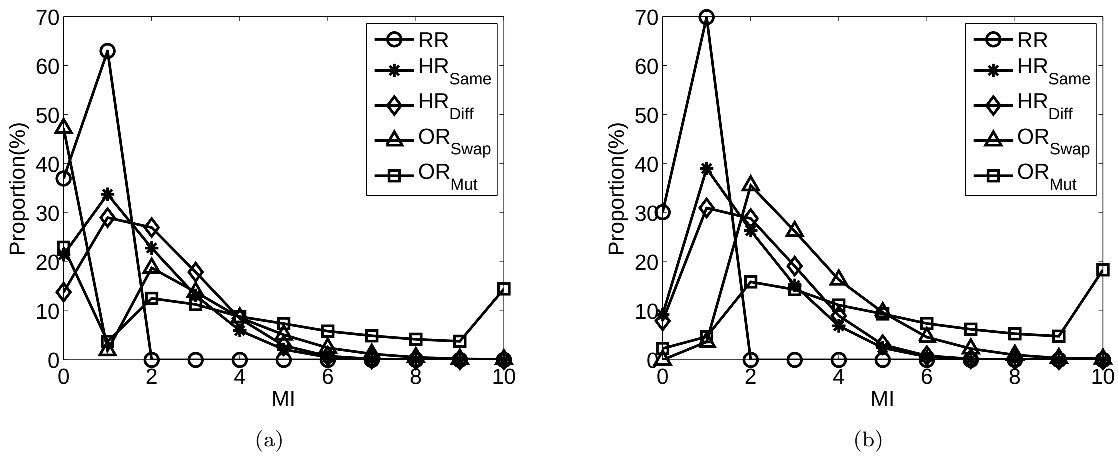


Figure 3.9: Empirical Distribution of (a) MI and (b) $MI > 0$ for 100,000 randomly generated chromosomes under different mutation operations.

Table 3.3: Statistical information on the MI distribution attained by the various mutation operators.

Mutation Operators	E(MI)	σ(MI)	E(MI|MI> 0)	σ(MI|MI> 0)
RR	0.6114	0.4256	0.6782	0.3945
HR_{Same}	1.7022	1.315	1.9676	1.2152
HR_{Diff}	1.9248	1.2813	2.0553	1.2186
OR_{Swap}	1.7267	1.9689	3.2749	1.5108
OR_{Mut}	4.4817	4.0682	5.6833	3.7628

RR attained noticeably lower MI^k over the various values of k, due to the nature of the mutation operation. While the rest of the mutation operators converged to the same $\text{E}(\text{MI}^k|\text{MI}^k > 0)$ for large k, OR_{mut} attained a significantly higher $\sigma(\text{MI}^k|\text{MI}^k > 0)$ relative to the various operators. Clearly from the empirical results, OR_{mut} consistently generate solutions that were further away from the original solutions, which will be beneficial in exploring the search space and providing genetic diversity within the evolving population. The rest of the mutation operators generate solutions that were closer to the original solutions, which was, on the other hand, more beneficial for local exploitation purposes.

Extending the analysis to the crossover operation, crossover innovation, CI, which measures the phenotype distance between an offspring and its phenotypically closer parent, is considered instead.

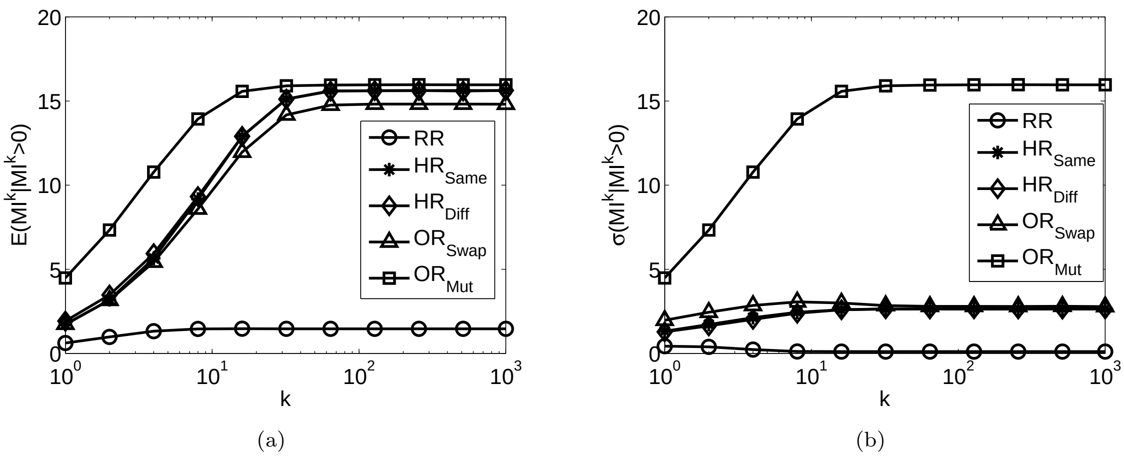


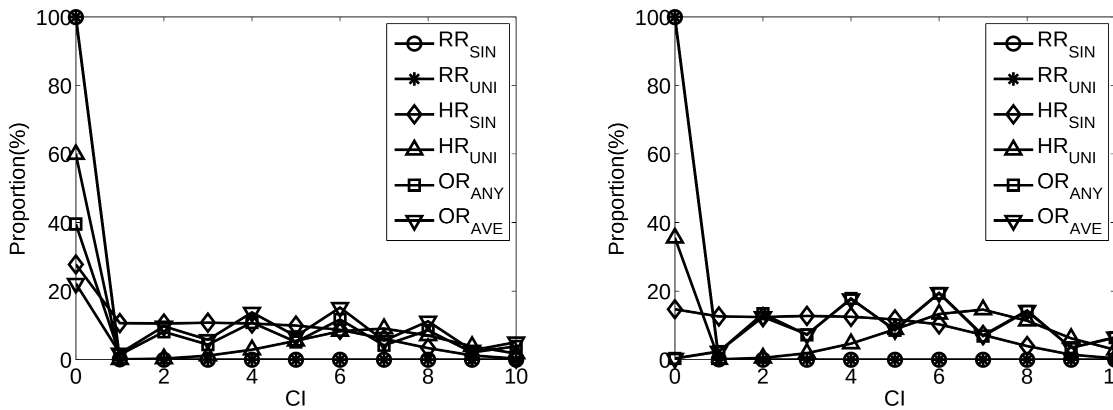
Figure 3.10: Empirical distributions of (a) $\mathrm{E}(\mathrm{MI}^k|\mathrm{MI}^k > 0)$ and (b)$\sigma(\mathrm{MI}^k|\mathrm{MI}^k > 0)$ over the number of mutation, k, for the various mutation operators

The expected value of CI strongly depends also on the original phenotype distance between the parents where a larger distance will most likely induce higher CI. The different crossover schemes for the various representations considered in this simulation study are summarized in Table 3.4. The crossover probability was arbitrarily set as 0.80. 100,000 pairs of chromosome were randomly generated and the various crossover operators were applied accordingly.

The resultant distribution of CI and CI>0 are plotted in Figure 3.11 and the corresponding statistical information are summarized in Table 3.5. As it is possible for the parent and offspring chromosomes to be identical even after the crossover operation due to neutral variation, the proportion of CI$\neq$0 was tabulated also. Again, as crossover operation of RR_{SIN} and RR_{UNI} only involve the varying of the weight vector, significantly lower CI was attained. On the other hand, even with the same crossover operators, HR_{SIN} and HR_{UNI} obtained higher CI due to the inclusion of the binary vector that allows the removal of assets from portfolio. Lastly, OR_{ANY} and OR_{AVE} have similar CI distribution after removing cases of neutral variation. Clearly, the imposition of the requirements to select the crossover point within viable assets reduced the occurrence of neutral variation by around 18%.

Table 3.4: Description of the various crossover operators in comparison.

Notation	Representation Schemes	Description of crossover operators
RR_{SIN}	Real Representation	Single Point Crossover i.e. given two parent chromosomes, a crossover point will be randomly selected and each chromosome will retain their original value (for the weight vector) before the crossover point with the remaining values swapped
RR_{UNI}	Real Representation	Uniform Crossover i.e. Given two parent chromosomes, each weight value will be randomly swapped with the corresponding weight value with a probability of pc
HR_{SIN}	Hybrid Representation	Single Point Crossover i.e. given two parent chromosomes, a crossover point will be randomly selected and each chromosome will retain their original value (for both the weight and binary vector) before the crossover point with the remaining values swapped
HR_{UNI}	Hybrid Representation	Uniform Crossover i.e. given two parent chromosomes, weight and binary value of an allele will be randomly swapped with the corresponding values of the other parent with a probability of pc
OR_{ANY}	Order-based Representation	Single Point Crossover as described earlier with crossover point randomly selected anywhere in the chromosome
OR_{AVE}	Order-based Representation	Single Point Crossover as described earlier with crossover point randomly selected within the mean portfolio sizes for the two parent chromosomes



(a) (b)

Figure 3.11: Empirical Distribution of CI (left) and CI> 0 (right) for 100,000 randomly generated chromosomes with different XO scheme

Table 3.5: Statistical information on the MI distribution attained by the various mutation operators.

Mutation Operators	E(CI)	σ(CI)	E(CI|CI> 0)	σ(CI|CI> 0)	Proportion of CI≠0
RR_{SIN}	0.1552	0.1246	0.1732	0.1192	89.59%
RR_{UNI}	0	0	0	0	54.33%
HR_{SIN}	3.0514	2.6629	3.6037	2.5267	84.67%
HR_{UNI}	2.6583	3.421	4.2716	3.4517	62.23%
OR_{ANY}	3.2452	3.2618	5.3507	2.5053	60.65%
OR_{AVE}	4.209	3.1371	5.3924	2.4953	78.05%

3.4 Local Search Operator

The design of global optimization techniques is governed by two competing goals, namely global reliability and local refinement [239]. The former is required to ensure that every region of the search space is covered to provide a reliable estimate of the global optimum and the latter to further improve the good solutions by concentrating search effort around their neighborhood. Many global optimization techniques achieved these two goals by adopting a combination of global and local search strategy.

Although EAs have shown general success in solving complex real-world optimization problems with various conflicting specifications, they suffer from slow convergence towards the optimum due to their strong stochastic bias [185, 95] and poor accuracy of the evolved solutions due to the lack of local refinement [141]. Memetic algorithms (MAs), motivated by the apparent need to employ both global and local search strategy to provide an effective global optimization method, extend EA by incorporating local search operators to complement the evolutionary operators. With this synergetic combination, the evolutionary operators will perform an adaptive, global sampling of the search space that actively generates solutions in new basins of attraction [104] throughout the evolution, while the local optimizers will efficiently refine these solutions and identify the corresponding local optimum within each basin. Studies have shown the relative advantages of MA over EA in terms of solution quality and computational efficiency [14, 176, 184, 185, 187, 259, 260].

Motivated by the compensatory property of EA and particle swarm optimizer (PSO) [30, 31, 121, 191], where the latter can enhance individuals generated by the evolutionary operators by sharing information between each other and their individually learned knowledge [121], a variety of MA that hybridize EA and PSO have been proposed in literature and applied widely in several real-world applications [31, 30, 98, 95, 96, 97, 121, 122, 123, 193, 199] and simulation results have verified the superiority of such MA over the lone applications of EA and PSO.

In general, EA and PSO can be hybridized in various approaches, for example, a simple two-phase approach where one algorithm is applied after the other [98, 193, 199] or interleavingly during the evolutionary search [95, 96, 97]. Alternatively, adopting one algorithm as the main evolutionary platform, the other can be encapsulated within, for example introducing the velocity operator into EA [31, 30, 121, 122, 123] or the mutation operator into PSO [77]. However, these approaches, which involve the application of EA and PSO interchangingly on the entire population/swarm during evolution, are only applicable for optimization problems that allow the same chromosome/particle representation under the two different algorithmic paradigms. Thus, they will have difficulty extending to optimization problems where the local optimizers operate in a reduced search space, hence requiring them to be applied independently to each candidate solution.

For this purpose, an EA-PSO hybrid model that specifically adopts PSO as a local optimizer to fine-tune each candidate solution individually will be introduced here. This memetic model will be especially suitable for the type of representation introduced earlier. Accounting for the high computational overheads of population-based optimization techniques, the candidate solution will be limited to the best solution found in the evolutionary search. The local optimizer will be triggered regularly during evolution for continual fine-tuning.

3.4.1 EA-PSO Memetic Models

As described earlier, the EA paradigm is based on basic concepts from the biological model of evolution, where the search dynamic of EA is driven by biologically inspired evolutionary operators like selection, crossover and mutation. The crossover and mutation operator manipulate and create potential solutions, while the selection operator provides the necessary convergence pressure. The strength of EA lies in their population-based search approach, which will generate higher diversity in the search space, reducing the likelihood to converge to the local optimum. Similar to EA, PSO is a form of population (swarm)-based optimization technique developed by Kennedy and Eberhart [128], which is inspired by the social behaviors of animals. In PSO, the position of the particles denotes candidate solutions to the optimization problem and their movements, influenced by its current position, memory and social knowledge of the swarm, are regarded as the search process for better solutions. EA and PSO possess compensatory property where the advantage of one algorithm can be the remedy for the other's weakness [31, 30, 121, 191].

PSO operates based on the social adaptation of information, where the memory of each particle allows knowledge of good solutions, i.e. its previous best position and the location of the global best solution, to be retained throughout the entire evolution. This allows constructive cooperation between particles, as particles in the swarm share information between them [121]. On the contrary, EA operates based on evolution from generation to generation (as opposed to single generation in PSO) by treating solutions from different generations distinctly. This results in valuable information being discarded at the end of generation as the algorithm starts almost randomly at each generation [31]. Particles of PSO will never be removed even if they are impossible to be the best solution, resulting in wastage of computational resources [31]. On the other hand, every individual in EA compete for survival where inferior parents are replaced by superior offspring at each generation. In addition, the crossover and mutation operator of EA generally helps in creating higher diversity in the evolving population as opposed to the velocity operator of PSO [30].

Empirical studies have revealed several characteristics about EA and PSO in general. [98] conducted comparative studies to investigate their differences, i.e. the fast convergence of PSO [244] and the reliability of global convergence of EA with enough generations, and suggested how they can be hybridized. Farsangi et al. [81] identified several parameter and problems setting that will influence their algorithmic performance.

A variety of EA-PSO hybrids have been proposed in literature. Grosan et al. [98] exploited the fast convergence of PSO and global reliability of EA and proposed the two-phase implementation of EA and PSO to solve geometrical place problems. Similar approaches have been adopted in the optimization of Profiled Corrugated Horn antenna [193, 199] and the simulation studies revealed that better algorithmic performance is achieved by first starting with PSO followed by EA. Alternatively, EA and PSO can be applied interchangingly during the evolutionary search by defining a hybridization coefficient that dictates the use of PSO and EA during the evolution [95, 96, 97]. Simulation results favored hybridization coefficients that are biased towards PSO, suggesting that the basic PSO can be significantly improved by using a small percentage of evolutionary operation during the evolution [96].

On the other hand, another type of EA-PSO hybrids adopts one of the algorithms as the main evolutionary platform and encapsulates the operators of the other algorithm within its search progress. For example, adopting EA as the main evolutionary platform, PSO can be applied to improve certain solutions in the population during the evolution, for example, the enhancement of the upper-half of the best performing solutions (elites) at every generation [30, 31, 121, 122, 123]. On the other hand, PSO could be used as the main evolutionary platform instead and evolutionary operators are added to complement the velocity operator, for example applying mutation on random particles chosen from the swarm [77] or introducing selection operation into the algorithm [6].

In the various memetic approaches discussed, EA and PSO could be implemented interchangingly during the optimization search, where the evolved offspring could be simply merged to form the particle swarm or parental chromosomes could be selected from the swarm of particles. However, this is only applicable for optimization problems that allow the same chromosome/particle representation under the two different algorithmic paradigms. Thus, such approaches will have difficulty extending to optimization problems where the local optimizers operate in a reduced search space, hence requiring them to be applied independently to each candidate solution.

Consider the portfolio management as an example, where a hybrid representation of binary and real-number vectors was adopted to respectively denote the inclusion of an asset within the portfolio and its corresponding proportion with respect to the available budget. While EA can be applied to simultaneously optimize the composition and weights of the portfolio, with PSO as a local optimizer to fine-tune the weight vector [48]. As each of the evolved portfolio has different binary vector, the different optimal weight vector, corresponding to the different portfolio composition, requires the local optimizer to be independently applied to candidate solutions. As such, the previous approaches of simply combining evolved offspring to form the particle swarm for the implementation of PSO is not applicable here as the local optimizer operators work on a reduced search space.

For this purpose, this paper proposes an EA-PSO hybrid model that specifically adopts PSO as a local optimizer to fine-tune each candidate solution individually. As for the candidate solutions, considering the high computational overheads of population-based optimization approaches, the memetic model limits the application of PSO to the best solution found in the evolutionary search. Since PSO operates on a population-based level, a particle swarm has to be generated from the candidate solution. Also, PSO will be invoked regularly during the evolution for the continual fine-tuning of the solutions.

Without any loss of generality, this particular MA, representing a hybridization of EA and PSO, with the latter used as a local optimizer for local fine tuning, will be denoted as EAPSO. It should be highlighted that the various operators in EAPSO has been deliberately made as generic as possible for the generality of the simulation results.

3.4.2 Knapsack Problem as a Proxy for Portfolio Optimization

The classical 0-1 knapsack problem is defined by a knapsack of capacity C and a set of N objects $O = \{o_1, ..., o_N\}$. Each of these objects, o_i has a value v_i and a weight α_i, where $i = 1, ..., N$. The problem amounts to selecting a subset $S \subseteq O$ of objects such that their combined value is maximal and the total weight does not exceed the knapsack capacity C. The mathematical description is as follows:

$$\max \sum_{i=1}^{N} b_i v_i \tag{3.6}$$

$$\text{s.t. } \sum_{i=1}^{N} b_i \alpha_i \leq C b_i \in \{0,1\}, i = 1, ..., N$$

where $b = \{b_1, ..., b_N\}$ is a binary vector that denotes the inclusion/exclusion of objects in the knapsack i.e. b_i=1 means that the i-th object is included in the knapsack and vice versa.

The fractional knapsack problem extends the original problem specifications by allowing the knapsack to be filled with fractional proportion of the objects. Each object will be available in a certain amount with different value per unit weight, $\overline{v_i} = \frac{v_i}{r_i}$. The problem is to fill the knapsack with the material mix, $\vec{x} = \{x_1, ..., x_N\}$ of maximal value, as stated in (3.6). However, this extension actually trivializes the problem as it can now be easily solved by the greedy algorithm where the object with the largest value per unit weight is being iteratively inserted into the knapsack [126] until it is full. The simplicity of the FKP can be attributed to the clear definition of the search space where the marginal value contribution of each objects is known.

$$\max \sum_{i=1}^{N} v_i x_i \tag{3.7}$$

$$\text{s.t. } \sum_{i=1}^{N} x_i \leq C, 0 \leq x_i \leq 1$$

In more general form, FKP (otherwise known as NFKP) can be defined as follows:

$$\max f(\vec{x}) \tag{3.8}$$

$$\text{s.t. } g(\vec{x}) \leq C, \vec{x} = \{x_1, ..., x_n\} \in \Re^N$$

where $f(\vec{x})$ and $g(\vec{x})$ are continuous functions. There is a huge variety of problem which can be construed based on this general definition [26], including portfolio optimization. For a clearer illustration, the problem definition of NFKP, and portfolio optimization are described below respectively,

- Selecting the appropriate objects to be included in the knapsack and determining their corresponding proportions.
- Selecting the appropriate assets to be included in the portfolio and determining their corresponding proportions.

Exploiting the structural similarity of these problems, a particular instance of NFKP will be formulated and implemented as the preliminary test platform for the proposed EA-PSO hybrid model. Such a test platform will allow the problem settings to be changed easily for the investigation of the algorithmic characteristics and provide insight to the algorithmic performance when extended to portfolio optimization later.

From the problem description, NFKP can be further decomposed into two smaller problems, namely determining which items to be included in the knapsack and their corresponding proportion. As such, apart from $\vec{x}$, a binary vector could be introduced to denote the inclusion and exclusion of the objects in the knapsack, so as to facilitate the removal and adding of objects to the knapsack. Similar hybrid representation has been adopted in portfolio optimization as discussed in the earlier sections.

The objective/fitness function for the proposed test problem is as follows:

$$\min F(\vec{x}, \vec{b}, \vec{x_G}, \vec{b_G}) = \sum_{i=1}^{N} b_{G,i} \times b_i \times \mid x_{G,i} - x_i \mid + \mid b_{G,i} - b_i \mid \tag{3.9}$$

$$\text{s.t. } \vec{x}, \vec{x_G} \in \Re^N, b_i, b_{G,i} \in \{0, 1\}$$

where $\vec{x_G}$ and $\vec{b_G}$ denotes the optimal weight and binary vector respectively. Essentially, the solution representation of NFKP is retained and the objective is to minimize the discrepancy between the optimized solutions, $(\vec{x}, \vec{b})$ and the global solution, $(\vec{x_G}, \vec{b_G})$. A fixed penalty (denoted by $\mid b_{G,i} - b_i \mid$) will be imposed when there is a mismatch of objects i.e. an object is present in the optimal combination but missing from the solution or vice versa and a variable penalty (denoted by $b_{G,i} \times b_i \times \mid x_{G,i} - x_i \mid$) proportional to the weight mismatch. The weight constraint was removed for generality. A new set of $(\vec{x_G}, \vec{b_G})$ will be randomly generated before every algorithmic run.

This is essentially a noisy problem where for a given solution $(\vec{x}, \vec{b})$, $\vec{x}$ might not be suitable for the corresponding $\vec{b}$, resulting in inaccurate fitness evaluation [256]. As such, the use of local optimizer in this case is actually to estimate the optimal $\vec{x}$ for each $\vec{b}$ for a more accurate fitness evaluation. Clearly, the various EA-PSO hybrid models reviewed are not applicable here due to the different $\vec{b}$ of each solutions, corresponding to different optimal $\vec{x}$.

3.4.3 Simulation Setup

The proposed EAPSO will adopt a hybrid representation, consisting of a binary vector,$\vec{b}$ and real vector,$\vec{x}$ of length, N to denote the inclusion of objects in the knapsack and their corresponding proportion. For the generation of the particle swarm, N_{local} $\vec{x}$ will be randomly generated by performing Gaussian mutation on every weight alleles of the candidate solution with $b_i = 1$, corresponding to their positions in the search space. A velocity vector, $\vec{v} = \{v_1, v_2, ..., v_n\}$ will be generated subsequently for each of the N_{local} particles, where v_i will be set to zero if $b_i = 0$, so as to keep the irrelevant weight vector unchanged during the PSO algorithm. It should be highlighted that the HR was considered over OR to enhance the generality of the results.

Unless otherwise stated, the parameter configuration of EAPSO in the simulation studies, including those in the subsequent sections, will follow Table 3.6. In all simulations, 30 independent runs were made. Also for a more accurate and fairer comparison, the same random seed was assigned to each set of runs to ensure that they started with the same initial population and possessed the same set of $\{\vec{x_G}, \vec{b_G}\}$.

Table 3.6: Algorithmic parameter settings of EAPSO for the simulation study.

General	Fitness Evaluation	500,000
	N	200
	Runs	30
EA (Global)	Population Size	100
	Archive Size	1
	Selection	Binary tournament selection
	Crossover	Single point crossover with probability 0.8
	Mutation	Bit flip mutation with probability $\frac{1}{N}$
PSO (Local)	Generation Interval, G_{local}	Variable
	Population size, N_{local}	Variable
	Time Step, T_{local}	Variable
	Inertia	0.7
	Individual weights	1.49
	Sociality weight	1.49

The inclusion of local search operation generally introduces additional computational cost in the aspect of fitness evaluations. As discussed earlier, two important design issues of local optimizer are their frequency and their application duration [104]. This is especially crucial in this case since PSO, being a population-based stochastic optimizer, has significant effects on computational load. Clearly, these issues are directly affected by the parameter setting of G_{local}, N_{local} and T_{local}. As a result, for this study, five different values for G_{local} were considered i.e. 20, 50, 100, 200 and 500. Specifically, G_{local} of 20 will mean that PSO will be applied at every 20 evolutionary generations of EAPSO. Each application of PSO will be limited to 2,500 fitness evaluations, which will be varied with different proportions between N_{local} and T_{local}. The different settings of G_{local}, N_{local} and T_{local} are shown in Table 3.7. EAPSO with $G_{local} = 0$ will operate exactly like an EA without any local optimizer.

Table 3.7: Different parameter settings for G_{local}, N_{local} and T_{local} and their corresponding index and notation.

Algorithm Index	Algorithm notation	G_{local}	N_{local}	T_{local}
1	EA	0	0	0
2	EAPSO20a	20	25	100
3	EAPSO20b	20	50	50
4	EAPSO20c	20	100	25
5	EAPSO50a	50	25	100
6	EAPSO50b	50	50	50
7	EAPSO50c	50	100	25
8	EAPSO100a	100	25	100
9	EAPSO100b	100	50	50
10	EAPSO100c	100	100	25
11	EAPSO200a	200	25	100
12	EAPSO200b	200	50	50
13	EAPSO200c	200	100	25
14	EAPSO500a	500	25	100
15	EAPSO500b	500	50	50
16	EAPSO500c	500	100	25

3.4.4 Simulation Result & Discussion

Figure 3.12 plots the fitness value attained under the various algorithmic configurations after 500,000 fitness evaluations, with zero being the optimal value. Clearly, the algorithmic performance is better for smaller N_{local} and will deteriorate instead for larger N_{local}. As such, given a limited number of fitness evaluation, it will be advisable to assign smaller N_{local} with longer T_{local} for PSO as a local optimizer. Different setting of G_{local} will also affect its efficacy, where larger G_{local} nullified its effect on the algorithmic performance of EAPSO, since PSO is not invoked as frequently to induce a significant impact on the evolutionary search. Conversely for smaller G_{local}, although the performance improvements for smaller N_{local} were significantly better, the performance deterioration for large N_{local} was amplified also, reflecting the high sensitivity of EAPSO to the parameter settings of N_{local} and T_{local}. Lastly, the deteriorative algorithmic performance observed for certain algorithmic configurations corresponded to the empirical claims in [104, 140], where without the correct application of the local optimizer, MA may not perform at its optimal level or worse, poorer than using EA alone.

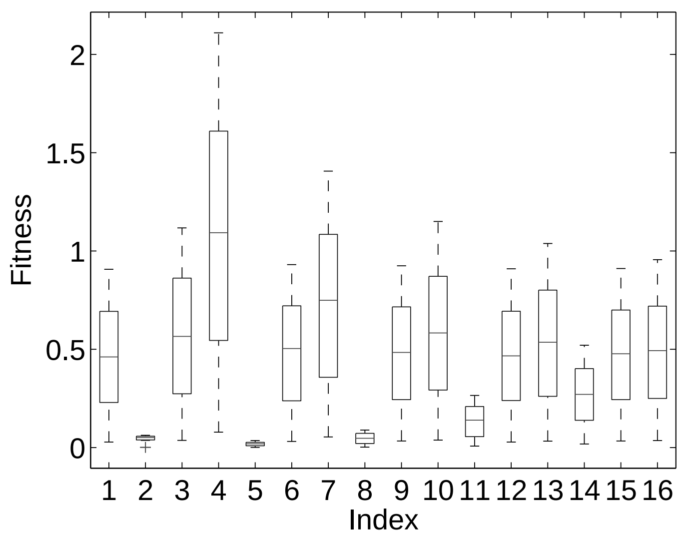


Figure 3.12: Fitness attained by the various algorithms after 500,000 fitness evaluation illustrated in box plots.

Figure 3.13 shows the evolutionary trace of the fitness function for three selected algorithmic configuration (EA, EAPSO20a and EAPSO100a) in one of the simulation runs. The evolutionary trace was analyzed in terms of fitness evaluations rather than the number of generations, as the former represents substantially how many times the fitness function has been evaluated by EAPSO in order to reach the optimization goals, which is more directly related to the computational load. To add more insight to the optimization process, the hamming distance between the global binary vector,$\vec{b_G}$ and that of the existing solution,$\vec{b}$ was included also to measure the discrepancy in the composition layout of the knapsack, whilst ignoring their proportion.

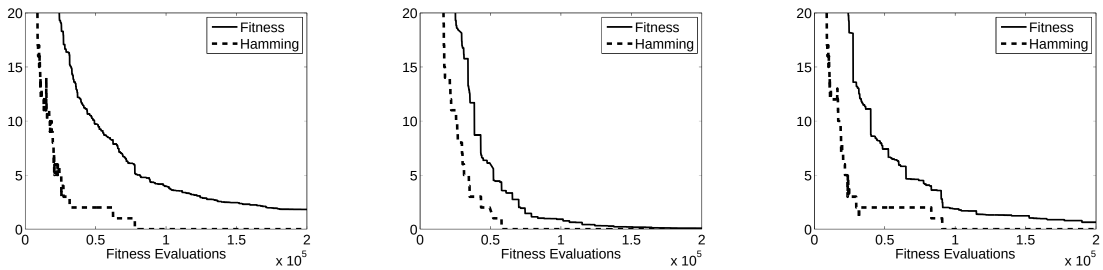


Figure 3.13: Evolutionary traces of the fitness and hamming distance for the best solution of (a)EA, (b) EAPSO20a and (c) EAPSO100a in one of the simulation run.

From the evolutionary traces in Figure 3.13(a), there is always a large discrepancy between the hamming distance and objective value which means that EA failed to identify the optimal weight vector associated with its current $\vec{b}$. However, the implementation of PSO for local fine tuning eliminates this discrepancy by rapidly identifying the optimal $\vec{w}$ for each $\vec{b}$ as shown in Figure 3.13(b) and 3.13(c). Although larger G_{local} will result in significant fitness improvement at each application as compared to smaller G_{local} as reflected by the sharper dips in the fitness trace, the latter allows immediate adjustment to any changes in $\vec{b}$.

To further investigate the algorithmic efficiency for different settings of G_{local}, the following algorithmic configurations were considered, namely EAPSO20a, EAPSO50a, EAPSO100a, EAPSO200a and EAPSO500a, and their mean fitness improvements over the 30 runs whenever the local search was triggered were plotted in Figure 3.14. Clearly, the marginal fitness improvement from PSO generally decreases as the evolutionary search progresses, most probably due to the general decline of the discrepancy between $\vec{x}$ and $\vec{b}$ as seen in Figure 3.13, which limits the improvements for PSO. Closer examination of Figure 3.14 reveals that the mean fitness improvement of EAPSO20a is high initially, but falls steeply to zero during the evolution, while there are still noticeable fitness improvements for bigger G_{local} even at the late stage of evolution.

Table 3.8 gives a more detailed comparison on the performance for the different settings of G_{local}, showing the mean objective values attained by the various algorithms, with the standard deviation shown in parentheses. Statistical test (ANOVA) revealed that the average objective value attained by EAPSO20a, EAPSO50a and EAPSO100a are statistically similar and significantly lower than the

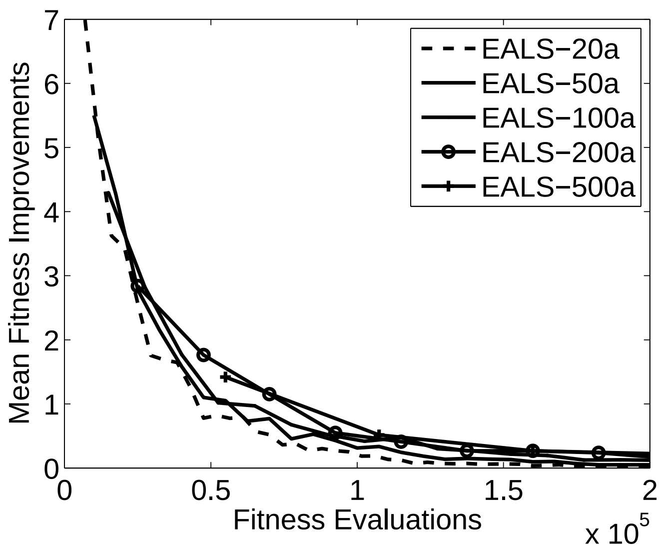


Figure 3.14: Mean fitness improvements whenever PSO was triggered at different fitness evaluations by EALS-20a, EALS-50a, EALS-100a and EALS-500a.

other two configurations, illustrating the robustness of the setting of G_{local}. Although EAPSO20a generated the highest fitness improvements amongst the different settings, these improvements are computationally inefficient as they are achieved via a significant amount of fitness evaluations. Instead, EAPSO200a generated higher improvements per fitness evaluation. Clearly, the correct setting of G_{local} plays an important role in balancing the local and genetic search. Over-utilization of local search will undermine the important role of diversity by the evolutionary operators [113], while under-utilization will limit the effectiveness of the local search to rectify the discrepancy between the binary and weight vector as reflected by the performance of EALS-500a.

3.4.5 Effects of Varying Problem Settings

The earlier results illustrated the algorithmic performance improvements of evolutionary search with the appropriate application of PSO for local fine-tuning. This section will assess the generality of the previous observations with further simulation studies involving varying problem specifications.

Generally, the problem difficulty will be affected by the number of available objects, N and the proportion of zero in $\vec{b_g}$, denoted as P_{zero}. As N directly controls the size of the search space, its

Table 3.8: Detailed performance comparison for different settings of G_{local}.

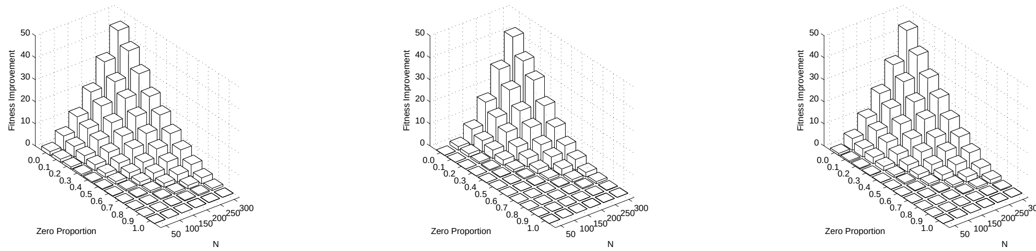
Algorithm notation	Objective Value Attained	Total Fitness Improvements	Fitness Evaluations	Fitness Improvements per fitness Evaluations	Percentage of Total Fitness Evaluations
EAPSO20a	0.0464 (0.0171)	36.0593	280000	1.29E-04	56%
EAPSO50a	0.0185 (0.0104)	23.792	167500	1.42E-04	34%
EAPSO100a	0.0468 (0.0278)	14.9145	100000	1.49E-04	20%
EAPSO200a	0.1369 (0.0817)	8.4172	55000	1.53E-04	11%
EAPSO500a	0.2699 (0.1509)	2.9206	23000	1.27E-04	5%

effect on the problem difficulty is rather intuitive. P_{zero} was set at 0.5 for the earlier simulations, which implies that 50% of $\vec{b_g}$, on average, will comprise of zero. In general, larger P_{zero} will result in optimal knapsack comprising of lesser objects and vice versa for smaller P_{zero}.

66 combinations of 6 values of N (i.e. $N = \{50, 100, 150, 200, 250, 300\}$ and 11 values of Pzero (i.e. $P_{zero} = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ were considered. Selected algorithmic configurations (EA, EAPSO20a and EAPSO200a) were applied to the test problem with different combinations of N and P_{zero} under the stopping criteria of 100,000 fitness evaluations.

Figure 3.15 shows the mean objective value attained under different problem settings, where a lower bar indicates lower objective value closer to the optimal value of zero. Clearly, the overall problem difficulty, reflected by the optimality of the solution attained under the same algorithmic configurations, increases with N and decreases with P_{zero}. This is rather intuitive since the former directly control the size of the search space while the latter affects the dimension of the weight vector to be optimized.

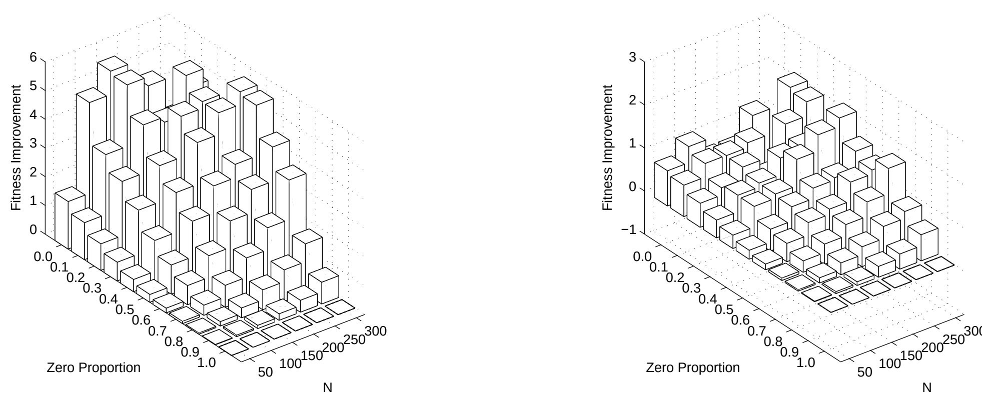
Figure 3.16 shows the mean fitness improvement attained at the different problem settings. There is a hill from the bottom-left corner to the top-right corner in the $P_{zero} - N$ plane, which marks the problem setting for greater algorithmic improvements by PSO. Basically, for large N



(a) (b) (c)

Figure 3.15: Mean fitness attained by (a) EA, (b) EAPSO20a and (c) EAPSO200a at different setting of N and P_{zero}.

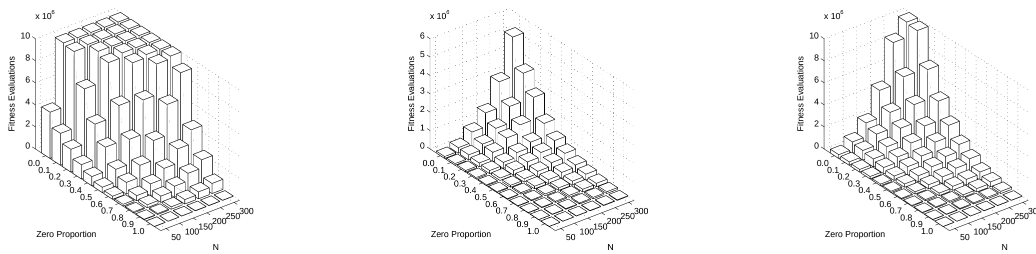
and small P_{zero}, the problem difficulty is higher due to the larger search space, resulting in limited improvements even when PSO is being applied. In fact, when $P_{zero} = 0$ and $N = 300$, EAPSO200a registered performance deterioration. On the other hand, for simpler problem settings of small N and large P_{zero}, the various algorithms were able to converge to the global optimum within the 100,000 fitness evaluations, resulting in minimal performance improvements.



(a) (b)

Figure 3.16: Mean fitness improvement for (a) EAPSO20a and (b) EAPSO200a with respect to EA at different setting of N and P_{zero}.

However, the importance of local optimizer in simple problems should not be under-estimated. Figure 3.17 plots the number of fitness evaluation required to reach with 5% of the optimal solutions for the various algorithmic configurations to assess their convergence speed, with the maximum fitness evaluation being capped at 10,000,000. The algorithmic improvements with the application of the local optimizer is clearly reflected in Figure 3.17 where EAPSO20a and EAPSO200a consistently requires lesser fitness evaluations to converge to the vicinity of the optimal solution across most of the problem settings as compared to EA.



(a) (b) (c)

Figure 3.17: Number of fitness evaluation required to reach within 5% of the optimal value for (a) EA, (b)EAPSO20a and (c) EAPSO200a at different setting of N and P_{zero}.

To put these results into context, the ratio of the fitness evaluation taken by EAPSO20a and EAPSO200a to that required by EA are calculated and shown in Figure 3.18. A ratio close to one will mean that the implementation of PSO does not affect the convergence time by much, while a value closer to zero will mean that EAPSO took only a small proportion of the computation time required by EA to converge. The low fitness evaluation ratio attained by EAPSO20a and EAPSO200a across the various problem setting is a clear indication of the algorithmic advantages of introducing PSO as a local fine tuning operator in evolutionary search. The relatively higher ratio obtained at the top corner is probably due to the implementation of the fitness evaluation cap, which prematurely end the algorithmic run of EA despite its sub-optimality. It should be highlighted in the case where P_{zero}=1, the problem essentially involves optimizing $\vec{b}$ only. As such, the implementation of PSO, which is in charge of optimizing $\vec{w}$, will only result in the wastage of unnecessary computation resource.

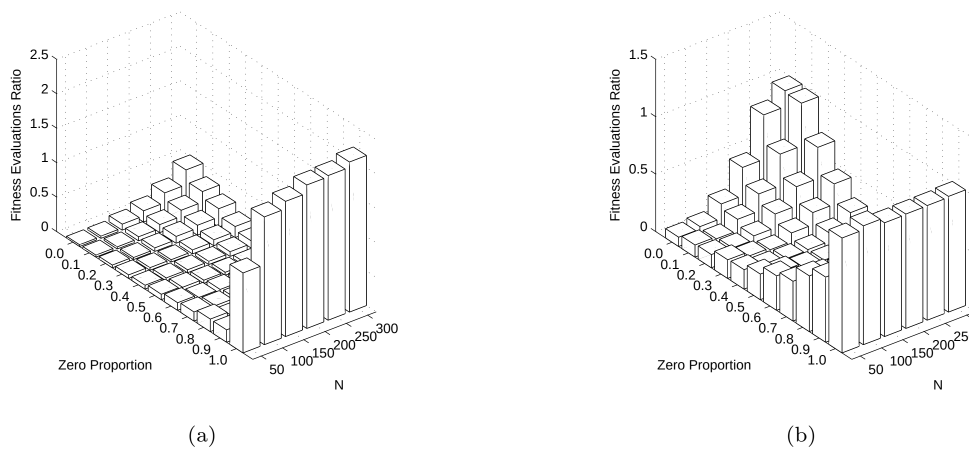


Figure 3.18: . Ratio of the number of fitness evaluation required to reach within 5% of the optimal solution for (a) EAPSO20a and (b) EAPSO200a to that required by EA at different setting of N and P_{zero}.

3.5 Dynamic Archiving Operator

3.5.1 Dynamic Optimization

Many real-world optimization problems involve complex and dynamic environments, where the objective functions, problem instance or constraints might vary with time, consequently altering the optimal solution. An example will be in portfolio management, where the portfolio must be continually monitored and rebalanced to adjust to the ever-changing market conditions. A problem of such nature is denoted as a dynamic optimization problem. In such a context, the optimization goal not only encompasses obtaining a good approximation of the optimal solution, but also adapting it to environmental changes over time.

In a certain sense, dynamic optimization problems can be regarded as the consecutive optimization of a series of different time-invariant optimization problems with the constraints of solving each problem before the next time instant of change. Hence, the most direct approach is to treat the problems in each time instant of change separately and restart the optimization process whenever changes in the problem environment are detected [194]. However, this approach demands the optimization techniques to have a high convergence speed, especially when the frequency of changes in the environment is high. Also, re-initialization of the evolving population constitutes a loss of genetic materials from the previous generations, which might be useful if the environment changes are not drastic. Lastly, this explicit restart approach assumes that any changes in the environment could be identified, which unfortunately might not always be the case [120] .

Reviewing the related literature works, several key algorithmic characteristics that should be present in the optimization techniques when handling dynamic optimization problems have been identified. There are, namely,

- High convergence speed.
- Sufficient genetic diversity to initiate a new search for the optimum.
- Ability to detect changes in the problem environment.
- Exploitation of past information whenever applicable.

These characteristics will enable the optimization techniques to possess the necessary flexibility to adapt accordingly to the time-varying optimum. Amongst the various optimization techniques available in literature, the adaptability nature of EA seems to be particularly suitable for such problems. In recent years, there have been growing interests in applying EA for dynamic optimization problems, motivating the development of several variants for this purpose [21, 22, 120, 175]. The following subsections will surveys the various related works in this area and subsequently proposed an archiving strategy that preserve useful information gained in the past, enabling the continuous adaptation of the underlying EA to the changing environment. The modified NFKP from the earlier section will be modified to a dynamic optimization problem, serving as the simulation testbed for the proposed method.

3.5.2 Handling Dynamism in Evolutionary Optimization

One of the problems facing traditional EA when solving dynamic optimization problems is that after the algorithm has converged to the existing optimum, it will have difficulty adapting to the new environment when a change occurs [254]. As such, it is necessary for the EA to maintain diversity throughout the evolutionary progress to adapt to the dynamic environment. Diversity here refers to the distribution of the solutions with respect to each other in the decision search space and/or the fitness landscape. Jin and Branke [120] classified the various approaches of dynamic evolutionary optimization into four different categories:

- Diversity Introduction: New genetic materials are explicitly added to the evolving population upon the detection of environment changes. Random restart represents the most direct example for this approach, where the evolving population is re-initialized whenever the environment changes. Alternatively, the mutation probability could be increased to generate more diverse solutions, for example, via hyper-mutation [50] or variable local search [242]. However, without proper archiving available, this introduced diversity might be at the expense of useful genetic materials of the previously fit individuals. The challenge here hence lies in introducing the appropriate amount of diversity. Specifically, too little will not be sufficient to improve the convergence speed, while too much might impede the evolutionary progress instead.

- Diversity maintenance: The basic idea is to maintain diversity throughout the evolutionary progress since a diverse population pool is instrumental for EA to adapt to environmental changes rapidly i.e. the probability of finding the solutions to a new problem is higher if the individuals are widely spread in the search space. One particular example is random immigrant approach [94], where randomly generated individuals are inserted into the evolving population to replace the existing individuals at specific intervals of the evolutionary progress. Sharing or crowding mechanisms (measured with respect to the decision search space and/or fitness landscape) [34] maintain diversity throughout the evolutionary progress. Nevertheless, the general consensus is that the continual focus on diversity during the evolutionary progress might impede evolutionary convergence.

- Explicit/Implicit Memory: If the optimum might repeatedly return to its pervious locations, a memory structure could be incorporated into the EA to provide information from its past generations. Generally, memory-based approaches could be classified into implicit and explicit memory. The former refers to the encoding of useful information from the past generations in the representation structure of the EA [99, 103, 147, 207]. Such approach is known as redundant representation as the chromosomal structure contains more information than necessary to define the phenotype. Contrarily, explicit memories adopt specific archiving strategies to store and retrieve useful information throughout the evolutionary progress.

- Multiple Populations: The basic idea in multiple populations is to simultaneously maintain presences in different regions of the search space, so as to detect or track any environment changes or emergence of new optimal region [20, 23, 241, 249]. Typically, at any time, a population will be exploiting the current optimal region while the rest of the population will be exploring the search space. Jin and Branke [120] described such a strategy as a kind of diverse, self-adaptive memory.

Though the various approaches differed in their general principles and implementation, their underlying motivation are inherently similar i.e. to provide EA with sufficient genetic diversity so that it can adapt to the new environment when a change occurs. While random restart stands at one end of the extreme of increasing the maximal diversity to the evolving population, this scheme does not significantly improve EA's capability in handling dynamic optimization problems. The explanation is that it is often hard for these randomly generated individuals or the more diverse solutions to establish themselves when the evolving population already contains highly fit individuals. Ideally, the individuals stored in the memory should be above-average optimality and well distributed across promising areas of the search space.

As such, the archiving strategy proposed here aims to simultaneously maximize genetic diversity (in the decision space) and minimize proximity with the current optimal solution (in the objective landscape). While multi-objective formulation is conventionally being utilized to address the inherent trade-offs between several objectives in optimization problems, multi-objective formulation is employed here as an archiving strategy to sustain a population pool that can maintain a tradeoff between the two stipulated objectives i.e. optimality and diversity. Simply said, the optimization problem tackled will remain a single-objective problem but the archive will adopt a multi-objective formulation. Such a formulation will prevent the archive from converging to the sole objective.

Besides deciding on the type of individuals to be stored, an archiving strategy should also comprise of replacement and retrieval strategies. As archive normally faced size constraint, replacement strategies dictates the manner in which the new individuals replaced the existing solutions. Possible approaches include deleting the individuals, which when deleted, will maximize the genetic diversity of the archive or removing older and poorer solutions by arbitrarily defining some ranking function. Here, the proposed archiving strategy will simply adopt crowding measure, specifically niche count, and remove solutions with higher niche count.

Retrieval strategies are concerned with how the archived solutions are being utilized during the evolutionary progress. There could be a continuous injection of archived solutions to the evolving population, which represents the diversity maintenance approach. However, such scheme might be detrimental to algorithmic convergence. As such, retrieval will only occur after the environment changes. The fitness of the best solution in terms of optimality will be re-evaluated at every generation and environment changes will be indicated by a deterioration of its fitness. Lastly, retrieval here denotes the merging of the evolving evolution and the archived solutions and selecting the survivors based on tournament selection.

3.5.3 Simulation Setup

To evaluate the feasibility of the proposed archiving strategy, the modified NFKP formulated earlier will be modified into a dynamic optimization problem. To simulate a dynamic environment, the problem landscape is periodically changed every τ generations during the evolutionary progress. The change here essentially refers to the application of the mutation operation on the global solution, $(\vec{x_G}, \vec{b_G})$. Apart from τ, another problem parameter α, is introduced here, which dictates the severity of the change at each instant. Specifically, the value of α, indicates the number of times in which $(\vec{x_G}, \vec{b_G})$ is mutated at each instant.

The algorithmic performance is measured by the average fitness of the best solutions at every generation.

$$f_{best} = \frac{1}{G} \sum_{t=1}^{G} f_{best,t} \tag{3.10}$$

where G is the maximal generation and $f_{best,t}$ is the fitness of the best solutions at time t. For each simulation, 30 independent runs were made. Also for a more accurate and fairer comparison, the same random seed was assigned to each set of runs to ensure that they started with the same initial population and possessed the same set of $(\vec{x_G}, \vec{b_G})$.

The evolutionary platform considered here will be similar to that in the earlier section, where a hybrid representation consisting of a binary vector, $\vec{b}$ and real vector, $\vec{x}$ of length, N to denote the inclusion of objects in the knapsack and their corresponding proportion is considered. Unless otherwise stated, the parameter configuration of EAPSO in the simulations, including those in the subsequent sections, will follow Table 3.9.

Table 3.9: Detailed performance comparison for different settings of G_{local}.

Fitness Evaluation	100,000
N	20
Runs	30
Population Size	100
Archive Size	1/100 with niche radius of 0.01
Selection	Binary tournament selection
Crossover	Single point crossover with probability 0.8
Mutation	Bit flip mutation with probability $\frac{1}{N}$

The proposed archiving strategy will be referred to as multi-objective memory (MOM) and will be compared with other test algorithms, as briefly summarized in Table 3.10. MOM will be evaluated against random restart (RR) and multi-objective archive (MOA), a modification of MOM where retrieval occurs at every generation. The two techniques can be categorized under diversity introduction and maintenance approaches respectively. The basic EA is included also to serve as the basis of comparison.

3.5.4 Simulation Result & Discussion

With τ and α respectively set at 100 generation (equivalent to 10,000 fitness evaluations) and 5, the evolutionary traces of the average $f_{best,t}$ in 30 simulations attained by each algorithm are plotted in Figure 3.19. In general,$f_{best,t}$ deteriorated abruptly at the stipulated generations but gradually improved as the algorithms adapted to the environment changes. Figure 3.19(a) illustrates the inherent self-adaptive capability of EA in dynamic environments, where even at the absence of dynamism-handling operators, reasonable algorithmic performance was attained. The external archive reduced

Table 3.10: Empirical values of the mutation innovation MI attained by the various mutation operators.

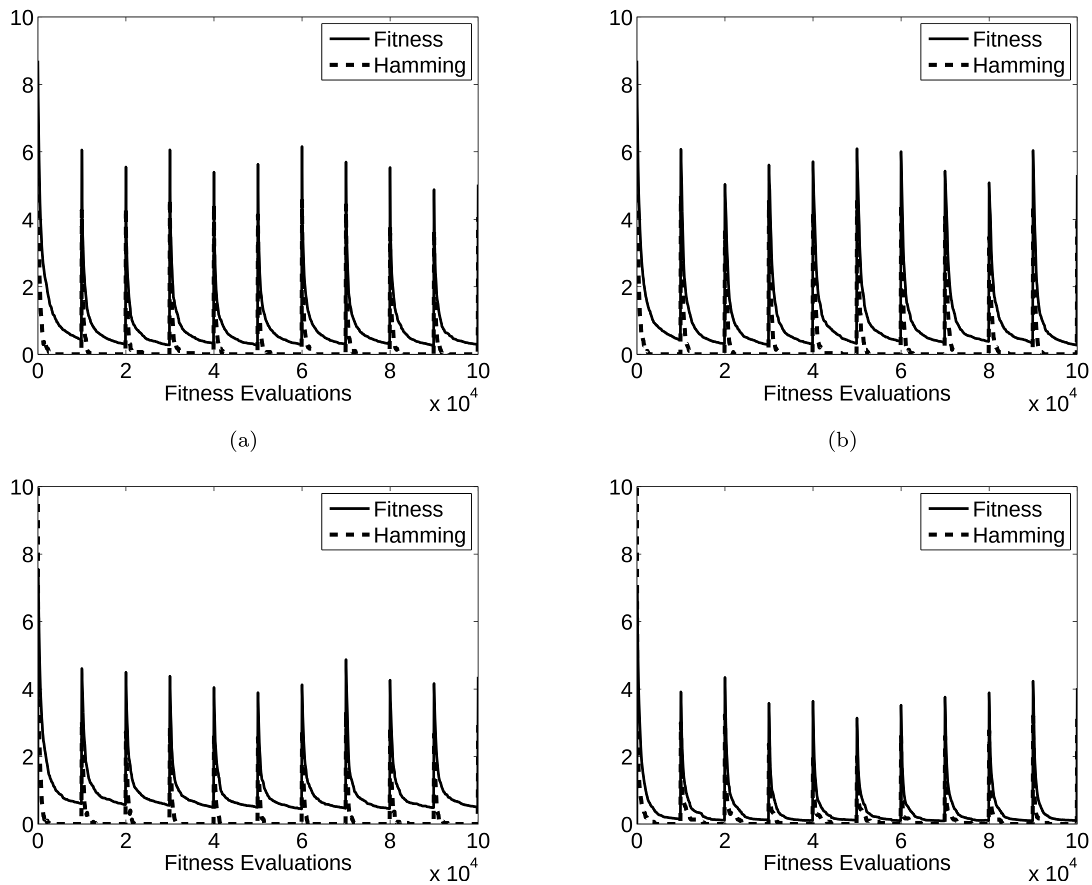
Notation	Techniques	Description of mutation schemes
EA	Standard EA	Archive of size one to store the solution with the highest fitness at every generation.
EA-RR	Standard EA with RR	Evolving population will be re-initialized whenever environmental changes are detected.
EA-MOA	Standard EA with MOA	Archive stores solutions of high optimality and diversity. Retrieval occurs at every generation.
EA-MOM	Standard EA with MOM	Archive stores solutions of high optimality and diversity. Retrieval occurs only when environmental changes are detected.

the sudden drop in fitness at every τ due to the enhanced genetic diversity provided. Comparing Figure 3.19(c) and 3.19(d), continual retrieval clearly impeded the evolutionary convergence process.

The distribution of the average $f_{best,t}$ and the hamming distance attained by the test algorithms are plotted in box-plots, as shown in Figure 3.20. Generally, the external archive enabled EA-MOA and EA-MOM to attain a lower average fitness (Hamming), implying that these two algorithms were able to track the combination relatively well. Statistical tests involving ANOVA and multiple comparison tests revealed that these differences are significant. Overall EA-MOM attained a significantly lower $f_{best,t}$ as compared to the other algorithms.

It will be instructive to analyze the evolutionary traces of $f_{best,t}$ in greater details to reveal further insight about their algorithmic characteristics. Figure 3.21 presents the closed-up illustration of the fitness traces of the various algorithms from generation 400 to 500. Clearly, fitness deteriorates significantly at the onset of the environment changes, but more severe for both EA and EA-RR. However, they managed to converge rapidly and eventually dip below EA-MOA. Overall EA-MOM remained robust to environment changes and was able to adapt rapidly to new environments.

Figure 3.22 compares the evolutionary traces of the average generic diversity in the evolving population for the various test algorithms in the same period. The corresponding trace of the archive for EA-MOA and EA-MOM was included also. The genetic diversity of EA remains stable



(a) (b) (c) (d)

Figure 3.19: Evolutionary traces of the average fitness and hamming distance for the best solution of (a) EA, (b) EA-RR, (c) EA-MOA and (d) EA-MOM in 30 simulations.

at the value of 1, except for the slight jumps after the environment changes. On the contrary, genetic diversity for EA-RR increased significantly at every τ due to the re-initialization of the evolving population, before settling back to 1 eventually. Despite so, EA-RR was not able to translate the enhanced genetic diversity to better algorithmic performance. Interestingly, although the archives of EA-MOA and EA-MOM maintained high diversity throughout the evolutionary progress, diversity in their evolving population was pronouncedly low, especially for the latter. Also, the change in diversity for EA-MOA at every τ is more gradual as compared to EA-RR. From fitness evaluation 500,000 to 520,000, diversity generally increased as the external archive was consistently introducing

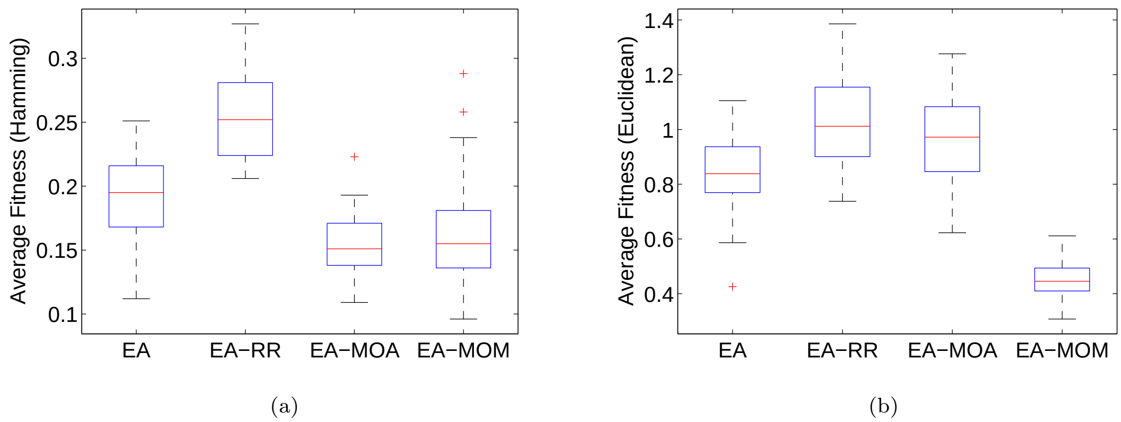


Figure 3.20: Box plot comparing the distribution of Hamming and Euclidean fitness of EA, EA-RR, EA-MOA and EA-MOM in 30 simulations.

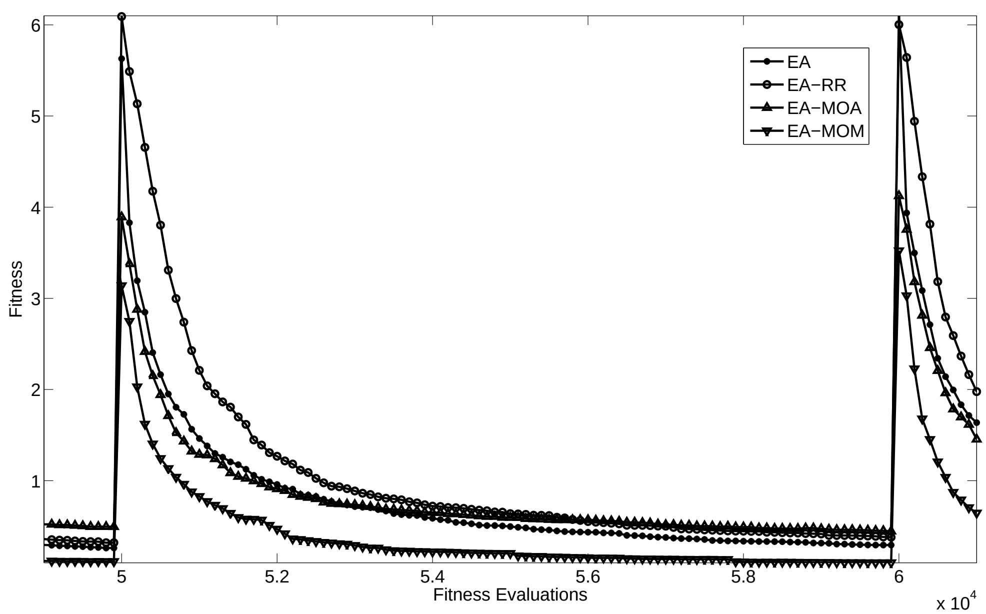


Figure 3.21: Evolutionary traces (closed-up illustration) of the average fitness for the best solution in the 30 simulations from generation 400 to 500.

fresh genetic materials into the evolving population. However, as the general fitness level in the evolving population improved, the archived solutions were compromised in the selection process, resulting in a gradual drop of diversity in the evolving population.

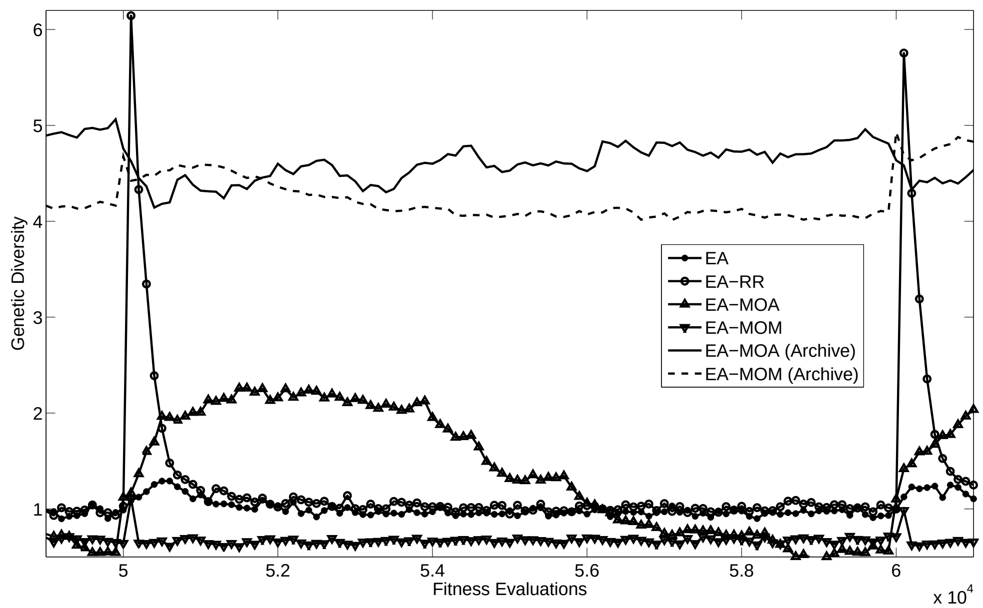


Figure 3.22: Evolutionary traces of the average genetic diversity in 30 simulations from generation 400 to 500 in the evolving population of EA, EA-RR, EA-MOA and EA-MOM and in the archive of EA-MOA and EA-MOM.

To examine the test algorithms under different dynamic environments, τ and α were varied. The set of values $\{1, 5, 10, 20, 50, 100, 200, 500, 1000\}$ were considered for these two problem parameters. The average $f_{best,t}$ attained by the test algorithms under the set of problem parameters is plotted in Figure 3.23. Same scales were chosen for the various plots to facilitate comparison. Intuitively, environments which change dramatically at a high frequency are hardest to track. As such, the worst algorithmic performance was generally at $\tau = 1$ and $\alpha = 1000$. Interestingly, the effect of τ on the algorithmic performance is more pronounced as compared to α, where the increase in τ triggered a larger improvement in $f_{best,t}$ attained as compared to a decrease in α. Also, the marginal fitness improvement diminished for large values of α and τ.

The algorithmic performance of EA, EA-RR, EA-MOA and EA-MOM can be compared easier

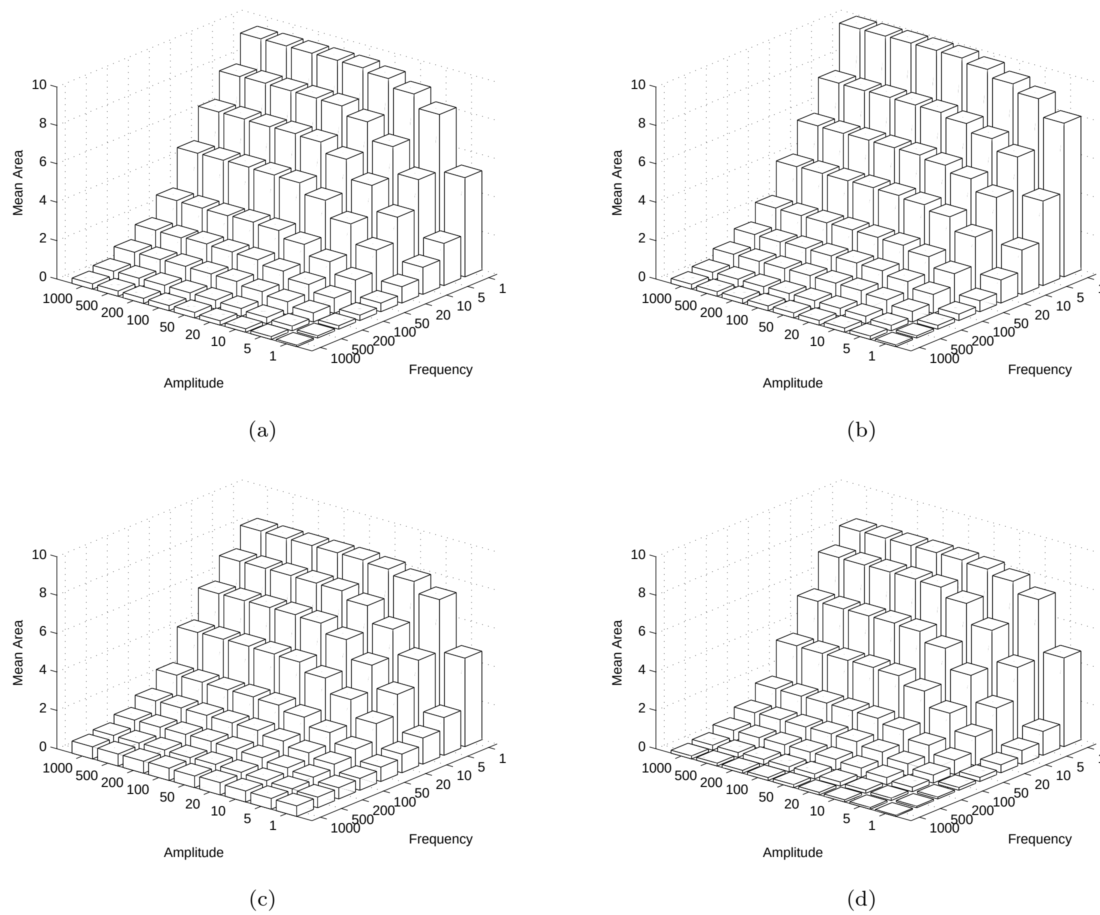


Figure 3.23: Mean area for (a) EA, (b) EA-RR, (c) EA-MOA and (d) EA-MOM at different setting of τ and α at the end of 500,000 fitness evaluations.

by analyzing the difference of their $f_{best,t}$ directly. The difference in $f_{best,t}$ of EA-RR over EA in the various problem settings were calculated and plotted in Figure 3.24. A positive difference, which corresponds to EA-RR having a higher $f_{best,t}$ than EA, indicates that EA-RR performed worse than EA in that particular setting. Also, as a mere difference in the average $f_{best,t}$ cannot be blindly regarded as performance difference between the algorithms due to the stochastic nature of the evolutionary platform, statistical tests should be included to improve credibility of the results. Particularly, the one-tailed t-test at a 0.05 level of significance was considered. It evaluates the viability of the null hypothesis, where the mean difference in $f_{best,t}$ between the two algorithms in comparison is purely stochastic. The result of the statistical analysis is shown in Table 3.11. The signs "+", "-" and "=" respectively denotes that EA-RR is significantly worse, significantly better and statistically indistinguishable relative to EA. The inclusion of RR actually worsened the capability of EA in handling dynamic environments for low values of τ and α. This is expected as EA-RR at $\tau = 1$ is analogous to solving an optimization problem with pure random search. Also, in slight environmental changes for low values of α, it will be better to exploit information from previous generations, rather than re-initializing a new population pool. Larger τ will negate the necessity of RR as there is ample time to adjust to the dynamic environments.

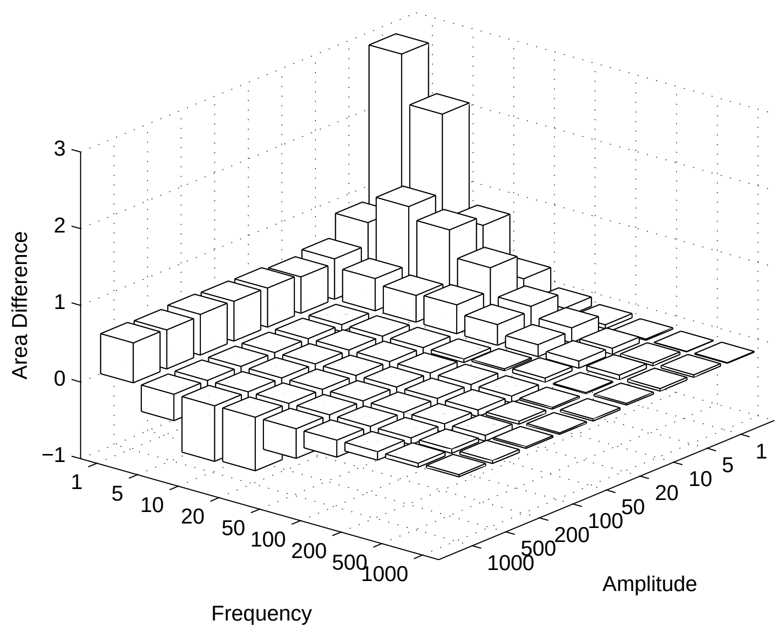


Figure 3.24: Difference of the mean area differences (random-normal) over 30 runs of normal archive versus MO archive. Positive difference indicates cases where area of normal archive is more than area of MO archive.

Putting the effects of the archiving strategy into perspective, Figure 3.25 compares the performance of EA-MOA and EA-MOM with respect to EA. In this case, a positive difference indicates that EA has a higher $f_{best,t}$ as compared to EA-MOA/EA-MOM. The corresponding statistical analysis is tabulated in Table 3.12 to 3.14. Clearly, EA-MOA only managed to outperform EA for smaller τ. This is expected as the continual injection of diversity to the evolving population in EA-MOA will

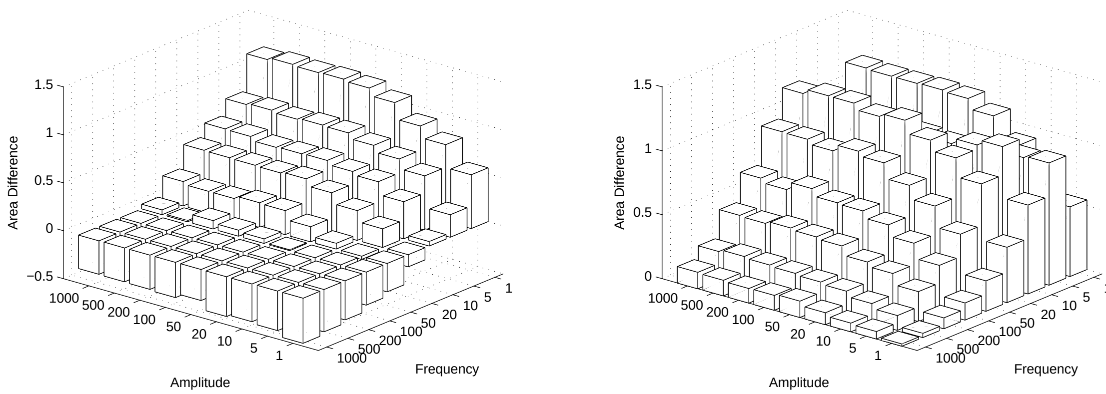
Table 3.11: Statistical Results (t-test at 0.05 significance level) of comparing EA over EA-RR in the various problem settings. The signs "+", "-"and "="respectively denotes EA is significantly better, significantly worse and statistically indistinguishable relative to EA-RR.

α	$\tau = 1$	$\tau = 5$	$\tau = 10$	$\tau = 20$	$\tau = 50$	$\tau = 100$	$\tau = 200$	$\tau = 500$	$\tau = 1000$
1	+	+	+	+	+	+	=	=	=
5	+	+	+	+	+	+	+	+	+
10	+	+	+	+	+	+	+	+	+
20	+	-	-	-	-	=	+	=	+
50	+	-	-	-	-	-	-	=	-
100	+	-	-	-	-	-	-	-	=
200	+	-	-	-	-	-	-	-	=
500	+	-	-	-	-	-	-	-	-
1000	+	-	-	-	-	-	-	-	-

indirectly impair algorithmic convergence. As such, for large τ where there are ample time to adjust to the environment changes, the self-adaptability of EA is sufficient to cope with the environment changes. Overall, EA-MOM consistently outperformed EA under the different problem settings, demonstrating the capability of the archiving strategy in providing genetic diversity throughout the evolutionary progress without impairing the convergence speed.

Table 3.12: Statistical Results (t-test at 0.05 significance level) of comparing EA-MOA over EA in the various problem settings. The signs "+", "-"and "="respectively denotes EA-MOA is significantly better, significantly worse and statistically indistinguishable relative to EA.

α	$\tau = 1$	$\tau = 5$	$\tau = 10$	$\tau = 20$	$\tau = 50$	$\tau = 100$	$\tau = 200$	$\tau = 500$	$\tau = 1000$
1	+	+	+	-	-	-	-	-	-
5	+	+	+	+	=	-	-	-	-
10	+	+	+	+	+	-	-	-	-
20	+	+	+	+	+	=	-	-	-
50	+	+	+	+	+	+	-	-	-
100	+	+	+	+	+	+	-	-	-
200	+	+	+	+	+	+	-	-	-
500	+	+	+	+	+	=	-	-	-
1000	+	+	+	+	+	+	-	-	-



(a) (b)

Figure 3.25: Difference of the mean area differences (random-normal) over 30 runs of normal archive versus MO archive. Positive difference indicates cases where area of normal archive is more than area of MO archive.

Table 3.13: Statistical Results (t-test at 0.05 significance level) of comparing EA-MOM over EA in the various problem settings. The signs "+", "-"and "="respectively denotes EA-MOM is significantly better, significantly worse and statistically indistinguishable relative to EA.

α	$\tau = 1$	$\tau = 5$	$\tau = 10$	$\tau = 20$	$\tau = 50$	$\tau = 100$	$\tau = 200$	$\tau = 500$	$\tau = 1000$
1	+	+	+	+	+	+	+	+	+
5	+	+	+	+	+	+	+	+	+
10	+	+	+	+	+	+	+	+	+
20	+	+	+	+	+	+	+	+	+
50	+	+	+	+	+	+	+	+	+
100	+	+	+	+	+	+	+	+	+
200	+	+	+	+	+	+	+	+	+
500	+	+	+	+	+	+	+	+	+
1000	+	+	+	+	+	+	+	+	+

Table 3.14 compares the algorithmic performance of EA-MOA and EA-MOM directly. At $\tau = 1$, there is no significant differences in their algorithmic performance, as EA-MOA , similar to EA-MOM in this case, referred to the archive at every generation due to the frequent changes. Overall EA-MOM outperformed EA-MOA over most of the problem settings except for the large values of α

at $\tau = 5$. This observation suggests that it might be beneficial to consider a retrieval strategy that refers to the archived solution less often under certain problem setting.

Table 3.14: Statistical Results (t-test at 0.05 significance level) of comparing EA-MOM over EA-MOA in the various problem settings. The signs "+", "-"and "="respectively denotes EA-MOM is significantly better, significantly worse and statistically indistinguishable relative to EA-MOA.

α	$\tau = 1$	$\tau = 5$	$\tau = 10$	$\tau = 20$	$\tau = 50$	$\tau = 100$	$\tau = 200$	$\tau = 500$	$\tau = 1000$
1	=	+	+	+	+	+	+	+	+
5	=	+	+	+	+	+	+	+	+
10	=	+	+	+	+	+	+	+	+
20	=	-	+	+	+	+	+	+	+
50	=	-	+	+	+	+	+	+	+
100	=	-	+	+	+	+	+	+	+
200	=	-	+	+	+	+	+	+	+
500	=	-	+	+	+	+	+	+	+
1000	=	-	+	+	+	+	+	+	+

3.6 Summary

This chapter discusses how the general MOEA can be extended for portfolio optimization. An order-based representation, which will be used in the evolutionary platform for subsequent chapters, was introduced. It main advantages lies in its better control in the type of asset forming the portfolio, allowing direct monitoring and control of the weight values for each asset at their point of inclusion. Flexibility of the proposed representation was exhibited also, where simple modifications to the initialization schemes allow the portfolio set of different pre-specified sizes to be formed.

Representation-specific variation operators were introduced subsequently. Preliminary empirical studies conducted compared their inherent algorithmic characteristics with other conventional variation operators under RR and HR. Generally, the proposed variation schemes attained higher MI and CI, which will be useful in the exploration of the search space and maintaining the genetic diversity throughout the evolutionary progress.

Lastly, a local search operator and an archiving strategy that could improve the algorithmic performance of MOEA were introduced in the last two sections. The former is essentially a PSO-EA hybrid model that can significantly improve efficacy and efficiency in algorithmic convergence. On the other hand, the latter represents a memory-based approach that enhances the capability of EA to deal with dynamic optimization problems. Details on their relevance to portfolio management will be apparent in the subsequent chapters where more advanced problem formulation are considered.

Chapter 4

Mean-Variance Analysis and Preference Handling

4.1 Introduction

This chapter is mainly devoted to the mean-variance analysis introduced by Markowitz [166]. Essentially, the central idea underlying the mean-variance model is that investors should not only be concerned with the realized returns, but also the risk involved with the asset holdings, measured by the standard deviation of the portfolio return in this case. As such, in the portfolio optimization process, the dual criteria of maximizing the expected returns and minimizing the associated risk should ideally be considered simultaneously, hence motivating the formulation of this problem in the domain of multi-objective optimization. This first part of this chapter will consider the fundamental mean-variance model and evaluate the feasibility of the proposed evolutionary multi-objective model, while the second part investigates the incorporation of preference criteria into the optimization process.

4.2 Markowitz Mean-Variance Model

In the Markowitz model, a perfect market without taxes and transactions costs was assumed, where short sales are disallowed and securities are infinitely divisible and can be traded in any (non-negative) fraction. Investors, assumed to be rational price-takers, will make their sole investment decision prior to the investment horizon in constructing a portfolio of assets, with their aim being to maximize their terminal wealth at the end of the period. The mathematical formulation of the Markowitz model is as follows:

$$\min f_1 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} \tag{4.1}$$

$$\min f_2 = \sum_{i=1}^{N} w_i \mu_i \tag{4.2}$$

subjected to

$$\sum_{i=1}^{N} w_i = 1 \tag{4.3}$$

$$0 \leq w_i \leq 1, i = 1, ..., N \tag{4.4}$$

where N is the number of assets available, μ_i is the expected return of asset i, σ_{ij} represents the covariance between assets i and j and w_i is the decision variable denoting the composition of asset i in the portfolio as a proportion of the total available capital. The goal in this optimization problem is to construct portfolios amongst the N assets i.e. finding the appropriate weight vector, $\vec{w}$, that can simultaneously satisfy the two conflicting objectives, namely, minimize the total variance (4.1), denoting the risk associated with the portfolio, whilst maximizing its expected returns (4.2). (4.3) gives the budget constraint for a feasible portfolio, while (4.4) requires all investment to be positive i.e. short selling is not allowed.

Essentially, the optimization problem is to find portfolios amongst the N assets that satisfy these two objectives simultaneously. As these two objectives are conflicting in nature, an optimal portfolio is one that has the maximum return with the minimum risk and the optimal solution in this case will comprise of a set of optimal portfolio illustrating the trade-off between these two objectives. This solution set, when plotted in the objective space will constitute the Pareto front, or more commonly known as efficient frontier in this context, as represented by FF in Figure 4.1.

Early related works were mainly single-objective approaches where one of the objectives is optimized, whilst pre-specifying a targeted value for the other objectives. However, the difficulty in specifying the targeted value is that it might not be within the efficient frontier. Alternatively, a

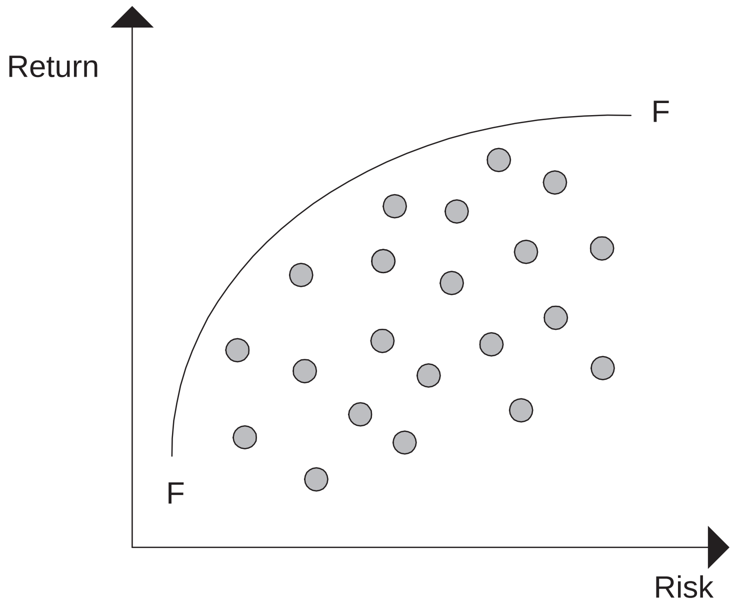


Figure 4.1: Illustration of the Efficient Frontier, FF

weighted function of the two objectives could be defined as such

$$\max \lambda f_1 - (1 - \lambda) f_2 \tag{4.5}$$

where different values of $\lambda \in [0, 1]$ will direct the search to different regions of the efficient frontier due to the varying emphasis on the portfolio's risk and expected return. An approximation of the efficient frontier can be obtained via repeated algorithmic iterations with different values of λ. This highlights the primary disadvantage of single-objective approaches, as repeated algorithmic iterations are required to obtain an approximation of the entire efficient frontier as opposed to a direct multi-objective formulation for this problem. Furthermore, a multi-objective search will result in a more diverse search of the decision space resulting in better algorithmic convergence, as demonstrated by the subsequent simulations.

By casting this problem into the domain of multi-objective optimization, more flexible problem formulation are allowed and the basic mean-variance model can be extended to consider more meaningful objective functions. Particularly, Arnone et al. [8] and Loraschi et al. [154] considered downside risk (i.e. distribution of the downside returns) in place of the return variance (1). Alternatively, additional objective functions have been incorporated to enhance the original model. To handle the cardinality constraints, Fieldsend et al. [82] considered the cardinal as an additional objective to be optimized. This approach allowed the direct extraction of the 2-dimensional cardinality constrained frontier for any particular cardinality. Other additional objectives considered in literature included surplus variance [229], portfolio value at risk [229], annual dividend [72] and asset ranking

4.3 Optimization Techniques for Portfolio Optimization

The main task in portfolio optimization is to select k out of N assets and optimize their weights so as to satisfy the objectives without violating the constraints. As demonstrated by Maringer [163], this is a NP-hard problem where the computational complexity scaled exponentially with the decision search space, for example the number of available assets, N or the size of the portfolio, K. As such, complete enumeration of the different possible decision variables is practically not feasible.

Analytic, closed-formed solutions, which are often desirable in optimization problems, are only available if simple modifications are applied to the Markowitz model, for example, omitting the non-negativity constraints from the basic model [17, 155, 237, 238]. As such, numerous optimization tools from the realm of Operations Research have been proposed, which include linear programming, quadratic programming, dynamic programming, greedy algorithm, gradient search and etc. These traditional approaches typically entail the transformation of the original problem into a single-objective problem and employs point-by-point algorithms. Even so the constraints set have to be modified to suit the optimization tools like network flow model [63], linear programming [135], interior point algorithm [231], 'branch and cut' approach [16], simplex model [250].

The major drawback with these methods is that they generally require the problem to be well-behaved, i.e. differentiability, satisfying the Kuhn-Tucker conditions or the problem specifics to be expressible under a certain format, as these techniques might not be proficient in handling the non-linear objectives and constraints functions within reasonable computational resources. In practical implementation, the portfolio optimization problems are often reformulated to suit the specifics of the optimization tools. As a consequence, their application is restricted to a limited set of problems and/or the model had to accept strong simplifications. This imposes a limit on the realism of the problem formulation as real-world situation usually garner a complex set of constraints and objectives.

These restrictions have motivated the development and application of evolutionary optimizers techniques like evolutionary algorithms, ant colony optimization [7], particle swarm optimization and etc. Amongst them, EA represents the more popular approach [251, 252, 40, 218, 35, 42]. Essentially, EA is implemented to optimize the asset composition in the portfolio and/or their corresponding proportion. Most of these works revolved around the Markowitz mean-variance model under a single-objective formulation, though slight modifications were made to improve the underlying model. In a separate study, Busetti compared EA performance with Tabu search and concluded that the former was better under that particular formulation. Similarly, PSO has been extensively applied on portfolio optimization and has been extended to improve its algorithmic performance [58, 139, 253, 41]. Similar to EA, single objective formulation of the portfolio optimization were considered namely minimize the risk with a targeted return value [253, 41], weighted approach [139], Sharpe ratio [58].

As discussed in the earlier chapters, their main advantages are its diverse search effort which will enhance the algorithmic convergence and its flexibility in implementation, where the evolutionary paradigm can be readily adapted to address the different problem specifics and constraints. The latter has resulted in works addressing more realistic constraints in portfolio optimization in the recent literature. However, it should be mentioned that the optimization process within polynomial time come at the expense of their optimality [222] , as the stochastic nature of such meta-heuristics could only guarantee feasible and near-optimum solutions. Nevertheless, in most real-world problems, speed is often of greater importance as compared to exact optimality.

Of more recent interest, there has been an increasingly number of works considering a multi-objective formulation of this problem, which represents a more direct approach given the underlying nature of the problem. Such formulation is more robust to the shape of the Pareto-front and approximation of the entire efficient frontier only requires one single iteration as opposed to conventional single-objective approaches. Also, such a formulation allows the consideration of more complex and sophisticated objective function. Specifically, in this chapter, a multi-objective evolutionary platform will be considered and applied to several test data. Subsequently in the second part of the chapter, preferences will be incorporated into the optimization process and it will be demonstrated how the evolutionary platform can be adapted as such.

4.4 Evolutionary Multi-Objective Portfolio Optimization

4.4.1 Simulation Setup

The evolutionary platform adopted was a generic elitist MOEA that maintained a fixed-size population and an archive to store the best solution discovered. Both the population and the archive are assigned a size of 100 each. The order-based representation proposed was adopted and the length of each chromosome depends on the number of assets available in each problem. In each generation, mating individuals were selected via binary tournament from the combined population of the existing evolved solutions and archive. The selection criterion was based on Pareto dominance. In the event of a tie, the niche count would be employed. Specifically, a niche radius of 0.01 in the normalized objective space was considered. The mechanism of niche sharing was used in the tournament selection as well as diversity maintenance in the archive. The mating individuals would subsequently undergo variation operation i.e. crossover probability of 0.8 and bit-wise mutation of $\frac{1}{N}$ to produce offspring for the next generation. The generational stopping criteria were varied for each problem based on their level of difficulty. Specifically, each problem was run sufficiently until their performance can be properly differentiated.

The proposed evolutionary model was applied to the basic portfolio optimization model to evaluate its basic feasibility. The various representation discussed earlier was considered also to investigate their algorithmic influence on the optimization process, namely the real vector representation [65], hybrid representation [225] and the order-based representation. The various algorithms configurations are described in Table 4.1. Prior investigations revealed that uniform crossover resulted in better algorithmic performance for the former two representations. As for the proposed order-based representation, the different initialization techniques mentioned earlier will be considered.

Table 4.1: Description of the various algorithmic configurations in the simulations for unconstrained portfolio optimization.

Algorithm Configurations	Notation
Real number representation with uniform crossover	RR
Hybrid representation with uniform crossover	HR
Order-based representation without W_{Max}	OR-1
Order-based representation with W_{Max} of 0.1	OR-2
Order-based representation with random W_{Max}	OR-3

4.4.2 Performance Metrics

Unlike single-objective optimization, there are several goals in multi-objective optimization [240], most notably proximity and diversity. The former describes the accuracy of the solution set while the latter measures how well the solution set is defined. Despite so, most of the studies in evolutionary portfolio optimization did not include diversity measures and statistical analysis that are commonly used in the performance assessment of multi-objective optimizers. While generational distance [221] and average relative distance to the efficient frontier [240, 75] have been used on separate occasion, other related works merely illustrate the efficient frontier attained [229, 7, 152].

In this paper, a set of proximity and diversity measures will be adopted that is commonly used in multi-objective optimization. The Generational Distance metric, GD was used to measure proximity. It quantifies how "far" the approximation of the efficient frontier found (EF$_{Known}$) is from the actual efficient frontier (EF$_{True}$)[243, 51] and is defined as

$$GD = (\frac{1}{m}\sum_{i=1}^{m} d_i^2)^{\frac{1}{2}} \tag{4.6}$$

where m is the number of solutions found, d_i is the Euclidean distance (in objective space) between the member i in EF$_{Known}$ and its nearest member of EF$_{True}$. A low GD signifies that EF$_{Known}$ is very close to the efficient frontier.

As for diversity, it depends on factors like the spread and spacing of the solution set. The former can be measured by the Maximum Spread, MS metric [261] which measures how well the efficient frontier is covered by EF_{Known} through the hyper-boxes formed by the extreme function values observed in both fronts. In order to normalize the metric, this metric is modified as

$$MS = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(\frac{\max_{i=1}^{m} f_m^i - \min_{i=1}^{m} f_m^i}{F_i^{max} - F_i^{min}})^2} \tag{4.7}$$

where f_l^i is the l-th objective of member i, F_i^{max} and F_i^{min} are the maximum and minimum of the l-th objective in EF_{Known} and L denotes the number of objectives. The greater the value of MS is, the more the area of EF_{True} is covered by EF_{Known}.

For the latter, the metric of spacing, S which measures how "evenly" solutions in EF_{Known} are distributed is chosen. It is defined as

$$S = \frac{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(d_i - \overline{d_i})^2}}{\overline{d}} \tag{4.8}$$

where d_i is the Euclidean distance (in objective space) between the member i and its nearest member in EF_{Known}. S will be low if the members in EF_{Known} are evenly distributed.

4.4.3 Simulation Result & Discussion

A set of portfolio optimization problems obtained from the OR-library [11] will be considered here to comprehensively evaluate the evolutionary model proposed. These problems contain the estimated returns and the covariance matrix for groups of assets in different stock market indices. Their details are summarized in Table 4.2. The difficulty of these problems is directly related to the number of assets available.

30 independent simulation runs were performed for all simulations and the same random seed was assigned to each set of the runs so that all algorithms start with the same initial population. The simulation results are illustrated by box plots in order to provide a statistical comparison of the performances for the various algorithms. Since a mere difference in the average of the qualitative metrics cannot be blindly regarded as performance difference between the algorithms, statistical test, namely the analysis of variance (ANOVA) is used to examine the significance of the mean difference between the various results.

Table 4.2: Description of Simulation Data Sets.

Problem Index	Data Source	Number of Assets
PORT1	Hong Kong, Hang Seng	31
PORT2	German, DAX 100	85
PORT3	British FTSE 100	89
PORT4	U.S. S&P 100	98
PORT5	Japanese Nikkei 225	225

Table 4.3: The average portfolio size and its corresponding standard deviation for the various solutions attained by the various algorithms in the different problems.

	RR	HR	OR-1	OR-2	OR-2
PORT1	30.99	4.74	3.30	4.38	3.52
	(0.031)	(0.48)	(0.35)	(0.62)	(0.28)
PORT2	84.98	15.68	4.64	9.77	7.21
	(0.043)	(2.67)	(0.82)	(1.08)	(1.50)
PORT3	88.98	16.84	4.27	9.23	7.02
	(0.042)	(2.97)	(0.74)	(1.67)	(0.91)
PORT4	97.99	24.09	6.15	13.60	12.08
	(0.0264)	(3.08)	(0.63)	(1.23)	(2.30)
PORT5	224.95	65.81	4.48	8.73	5.65
	(0.089)	(11.16)	(0.57)	(1.19)	(1.24)

The boxplots of the various performance metrics are illustrated in Figure 4.2. Varying stopping criteria are considered for different problems to compensate for their relative difficulty i.e. maximum generations of 100 for PORT1, 300 for PORT2, PORT3 and PORT4 and 500 for PORT5. The performance of RR was significantly poorer than the rest, especially in terms of diversity of the EF_{Known} attained, as reflected by their low values of MS. This was due to the nature of the representation, which favored large portfolio sizes that were near to N, as verified by Table 4.3. As such, EF_{Known} for RR was limited to the region of low return and risk due to excessive diversification and thus failed to cover the entire efficient frontier. This is illustrated in Figure 4.3, which depicts EF_{Known} obtained by RR in PORT4.

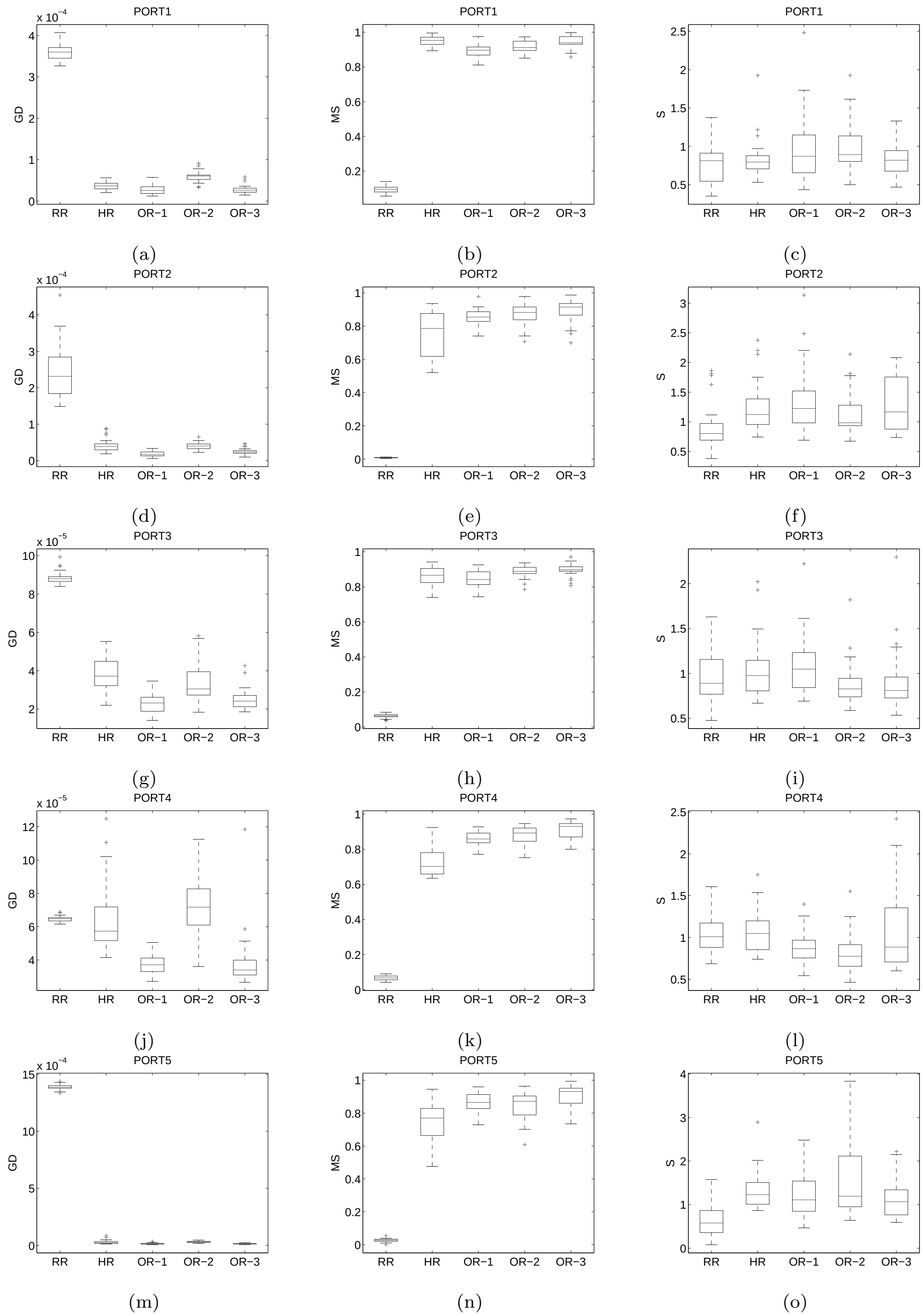


Figure 4.2: Box plots illustrating GD, MS and S obtained under the different algorithms for the different problems with varying stopping criteria.

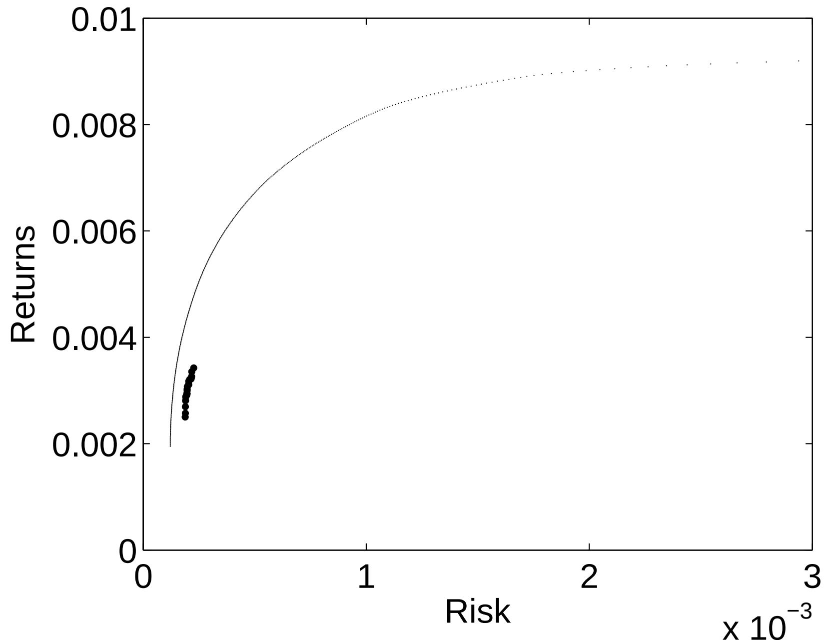
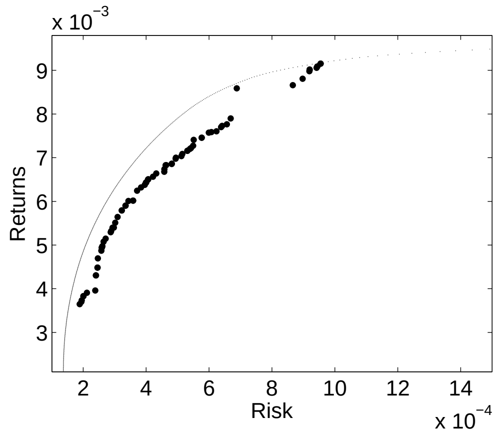


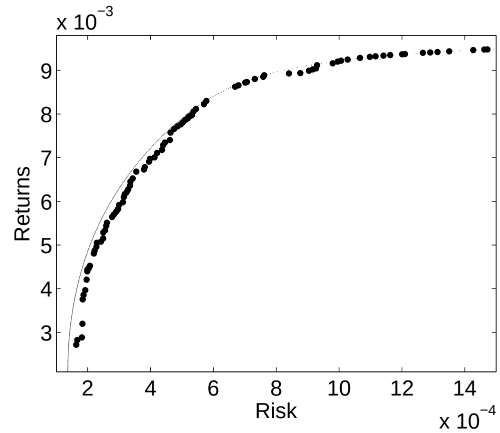
Figure 4.3: EF_{Known} of PORT4 obtained by RR in one of the algorithmic runs, with the corresponding EF_{True} denoted by the dotted-line.

ANOVA tests revealed no significant differences between the GD attained by HR and OR-3. However, there were significant differences in the degree in which they satisfied the diversity goal of attaining a solution set that spans the entire EF_{Ttue}. Except for PORT1 and PORT3, the ANOVA test revealed that OR-3 actually attained a significantly higher MS as compared to HR. Figure 4.4 and 4.7 compare the EF_{Known} obtained by HR and OR-3 in PORT2 and PORT4 and they clearly illustrate the difference in diversity under these two representations, in accordance with the box plots. It is evident from Figure 4.6 that OR-3 was able to attain a set of solutions that is close to EF_{Ttue} with sufficient level of diversity for the rest of the problems. However, it is noticeable that certain regions of EF_{Ttue} were not well-defined. Hence to further improve the algorithmic performance of OR-3 in terms of diversity, local search operators could be deployed in future works to improve the algorithmic convergence.

A closer examination of the box plots in Figure 4.2 reveals differences in the algorithmic performance for the various initialization techniques. As discussed earlier, OR-1 will favor smaller portfolios, thus the algorithm will work with fewer assets initially and then gradually increases the

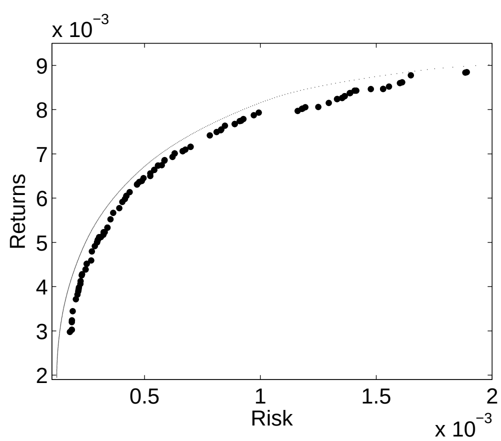


(a)

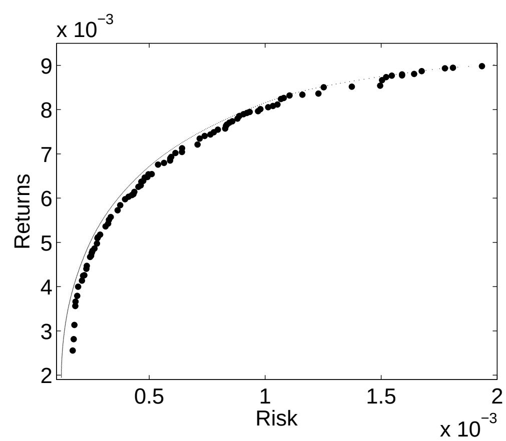


(b)

Figure 4.4: EF_{Known} obtained by (a)HR and (b)OR-3 for PORT2 in one of the algorithmic runs, with the corresponding EF_{True} denoted by the dotted-line.

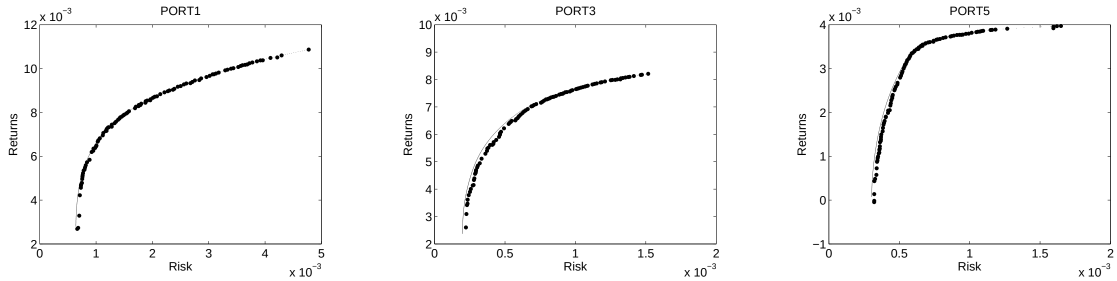


(a)



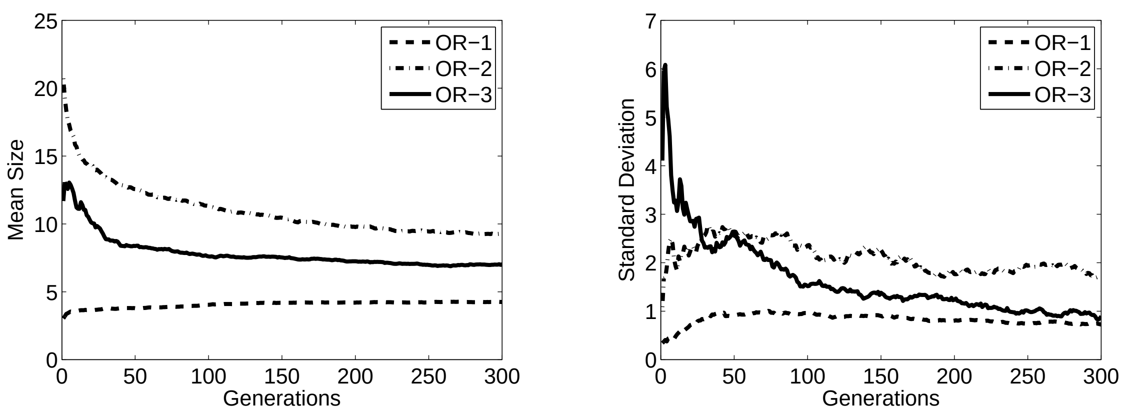
(b)

Figure 4.5: EF_{Known} obtained by (a)HR and (b)OR-3 for PORT4 in one of the algorithmic runs, with the corresponding EF_{True} denoted by the dotted-line.



(a) (b) (c)

Figure 4.6: EF_{Known} obtained by OR3 for (a)PORT1, (b)PORT3 and (c)PORT4 in one of the algorithmic runs, with the corresponding EF_{True} denoted by the dotted-line.

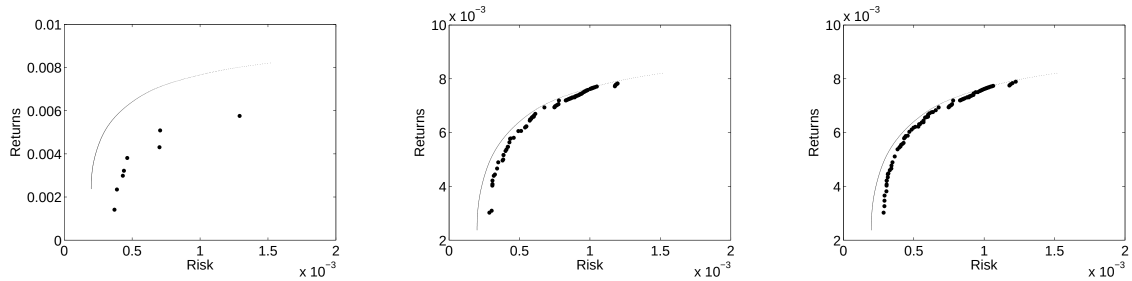


(a) (b)

Figure 4.7: Evolutionary trace of the (a) average portfolio sizes and the (b) corresponding standard deviation in PORT3 for three different algorithms i.e. OR1, OR2 and OR3.

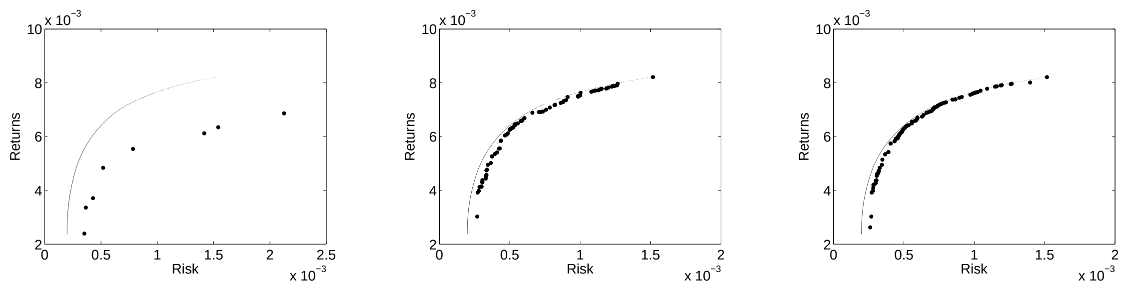
portfolio size during the evolutionary search progress. This can be observed from the evolutionary traces of the portfolio sizes in Figure 4.7. On the other hand, the application of the fixed initialization limits of 0.1 or random initialization increased the initial portfolio sizes, with the latter providing more diversity in the initial population, as observed in Figure 4.7(b).

The importance of diversity in the initial population is reflected in the evolutionary traces of the objective space. The diverse initial population generated by OR-3 as compared to OR-1 (Figure 4.8(a) versus 4.9(a)) resulted in a more diverse set of solutions (in terms of MS) being evolved eventually at generation 100.



(a) (b) (c)

Figure 4.8: EF_{Known} attained by OR1 at different generation i.e. (a) generation 0, (b) generation 50 and (c) generation 100 in PORT3, with the corresponding EF_{Ttue} denoted by the dotted-line.



(a) (b) (c)

Figure 4.9: EF_{Known} attained by OR3 at different generation i.e. (a) generation 0, (b) generation 50 and (c) generation 100 in PORT3, with the corresponding EF_{Ttue} denoted by the dotted-line.

4.5 Handling Preferences in Portfolio Optimization

4.5.1 Preferences in Multi-Objective Optimization

The MOO process broadly comprises of three main stages, namely measurement, search and decision making [52]. Most work related to EMO tends to focus on the second aspect, while the first aspect is addressed once the underlying formulation of the MOP can be mathematically defined. Decision making, on the other hand, is less studied since most of the time it is unnecessary, as it is implicit in the search process itself. MOO are concerned with problems of multiple conflicting objectives and in the absence of information regarding the relative importance between them, a set of Pareto optimal solution will ultimately be yield where any improvements in one objectives can only be obtained at the expense of degradation of the other objectives [83]. Ultimately, the decision maker has to select one single solution (or a particular region) from the Pareto optimal set, depending critically on his preferences towards the various objectives and/or the solution characteristics specifics i.e. robustness of the solution.

According to Horn [109], preferences can be specified a prior (before the search process), a posteriori (after the search process) or interactively (during the search process) to the evolutionary paradigm. Preferences towards different objective functions can be implemented by a weighting /utility function that combines all objectives into a single objective function. Alternatively, dominance relationship can be redefined in the case of ϵ-dominance [145], α-dominance [112] and s-dominance [208] to incorporate users' preferences into the dominance ranking. The emphasis on the various objectives in accordance to the weights will drive the algorithmic convergence to the appropriate region.

Alternatively, the decision maker could be particularly interested in certain regions of the Pareto front. In related works, such preferences are generally expressed in the form of reference points, where the objective is to reach a Pareto front region located near them. Deb et al. [61] proposed a Euclidean distance measure which acts as a secondary selection criterion to promote solutions that are nearer to the pre-specified reference point. Subsequently, a weighted stress function method was proposed which allows the decision maker to control the dispersion of the solution around the targeted region [83]. These two methods represent a prior approaches which require the definition of the reference vector before the algorithmic run. An a posteriori approach was proposed by Miettinen [172] where a weighted metrics is used to select solutions that are closer to the ideal criteria vector after the optimization process. The problem with reference vector is that the decision maker needs to have certain knowledge about the best solutions or regions of the Pareto front, which could be hard, especially for a prior approaches.

4.5.2 Capital Asset Pricing Model

Even though knowledge of the efficient frontier is important, often, portfolio managers are only interested in specific regions or points along the efficient frontier in practical situations, for example the efficient portfolio as described in the capital asset pricing model (CAPM). The key principles underlying CAPM are illustrated in Figure 4.10. The point, R_f represents the risk-free return available in the market to an individual, for example through government treasury bills. Line CC denotes the capital market line, which is a straight line that passes through R_f and is tangential to the efficient frontier, FF. The point, a, at the intersection of CC and FF is the efficient portfolio. The significance of the efficient portfolio is that any combination of it and the risk-free asset, attainable by either lending or borrowing at the rate of R_f, will allow the individual to operate any point on the capital market line and above the efficient frontier, resulting in higher returns for any given amount of risk than any portfolio of risky assets on FF.

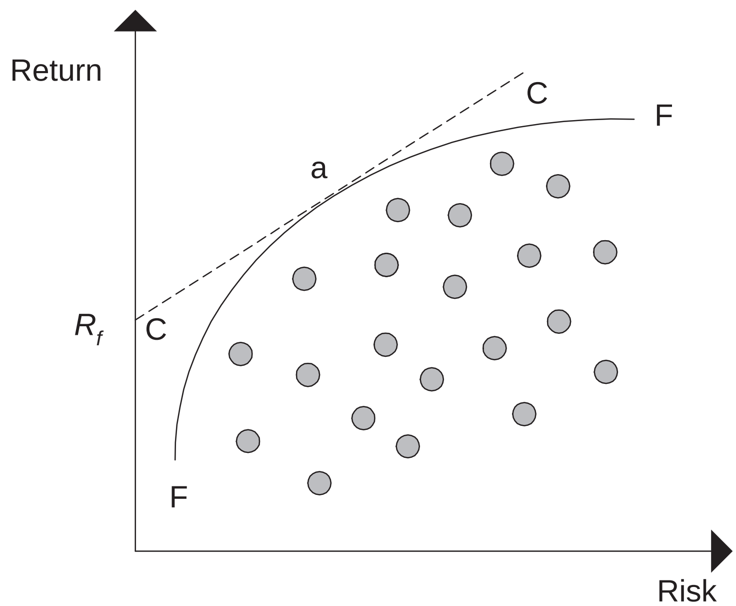


Figure 4.10: Illustration of the CAPM.

Mathematically, the efficient portfolio is the point on the efficient frontier that can maximize the objective function as follows [49]:

$$\max F_3 = \frac{F_2 - R_f}{F_1} \tag{4.9}$$

where R_f is the risk-free rate. Essentially, the problem is to maximize the gradient of the tangential line to the efficient frontier originating from the risk-free rate. This value is otherwise, more commonly known as the Sharpe ratio which measures the excess return (or Risk Premium) per unit of risk in an investment asset or a trading strategy [216].

Contrary to the portfolio optimization problem introduced earlier, the optimization problem of finding the efficient portfolio essentially combined the original two objectives into a single objective function. While the direct approach is to formulate this as a single-objective problem, this chapter considered the third objective as a preference criterion in the original dual-objective problem model i.e. (4.1) and (4.2) being the main objectives while (4.9) denotes the secondary criterion. The motivation for such a formulation is that the original multi-objective formulation will maintain the genetic diversity within the evolving population while the secondary objective will drive the algorithmic convergence towards the preferred region. Similar to the previous a prior approaches, the additional objective will be regarded as a secondary criterion in the selection process after the optimality comparison. To provide the additional stimulus, the PSO local search operator will be incorporated into the evolutionary platform, focusing on improving solutions with respect to the secondary criterion.

4.5.3 Simulation Setup

Three different evolutionary platforms were considered here so as to evaluate the feasibility of the proposed memetic model in handling preference criterion in multi-objective portfolio optimization. The first evolutionary platform adopted was identical to the one used in section 3.4 (single objective version), adopting the hybrid representation of a binary vector, $\vec{b}$ and real vector, $\vec{x}$ to denote the inclusion/exclusion of each individual assets in the portfolio and their corresponding weights. Algorithmic configuration adopted was as per the earlier simulations and the objective is to search for the efficient portfolio (4.9). To satisfy the budget constraint (4.3), the weights of the relevant assets in the portfolio will be normalized to one before the fitness evaluation.

The second EA extended the above-mentioned EA into the multi-objective domain, by incorporating Pareto-based ranking technique and diversity preservation mechanism. The selection criterion was based on Pareto ranking and in the event of a tie, the niche count was employed. The mechanism of niche sharing was used in the tournament selection as well as diversity maintenance in the archive. Both objectives (4.1) and (4.2) were simultaneously considered. The archive size was expanded to 100 to store the non-dominated solutions obtained during the algorithmic run.

Lastly, for the third EA, preference was incorporated into the optimization process of the previous multi-objective EA, by simply adding (4.9) into the selection process after the optimality criteria [49]. This allowed the evolutionary search to be focused on the preferred regions, resulting in the solutions converging nearer to the efficient portfolio, hence providing more alternatives around the targeted region.

In the various evolutionary approaches, PSO, being a single objective approach, will optimize (4.9) solely. While the candidate solution is obvious for the first case since there is only one solution in the archive at any time, the candidate solution for the latter two cases will be the solution with the maximum value for (4.9) in the archive.

The algorithmic index, notations and descriptions of the various evolutionary approaches are summarized in Table 4.4. Following previous simulation results that advocates lower N_{local} with respect to T_{local} , two algorithmic configurations, $\{G_{local}, N_{local}, T_{local}\}$ were considered here i.e. {20, 25, 100} and {100, 25, 100}. For brevity, simulation result of varying N_{local} and T_{local} were not considered here though simulations do revealed similar observations as the previous studies.

The various algorithms were applied to a set of portfolio optimization problems obtained from the OR-library [11], which contains the estimated returns and the covariance matrix for groups of assets in different stock market indices. The details of the test problems are summarized in Table 4.2. Three different R_f were considered for each problems, namely the 25%, 50% and 75% percentile of the returns, which corresponded to different efficient portfolios. The various values are listed in Table 4.5 and Figure 4.11 gives a clear illustration of how these values are obtained in each problem.

4.5.4 Simulation Result & Discussion

The algorithmic performances were evaluated based on the number of fitness evaluations required to reach within 5% of the optimal fitness for the respective problem and R_f. The mean fitness evaluations taken in 30 runs are illustrated in boxplots, as shown in Figure 4.12-4.16. In most cases, the MO approach located the preferred region since the efficient portfolio is ultimately part of the efficient frontier. However, there were improvements in the convergence time to obtain solutions in the vicinity of the preferred region after introducing the preference knowledge in the selection criteria. Interestingly, these improvements were especially significant whenever there was a large

Table 4.4: Description of the various algorithms configurations considered in the simulation for portfolio optimization.

Algorithm Index	Algorithm notation	Description
1	SO	Single-objective EA
2	SO-LS20	Single-objective EA with PSO applied at every 20 generations
3	SO-LS100	Single-objective EA with PSO applied at every 100 generations
4	MO	Multi-objective EA
5	MO-LS20	Multi-objective EA with PSO applied at every 20 generations
6	MO-LS100	Multi-objective EA with PSO applied at every 100 generations
7	pMO	Preference-based Multi-objective EA
8	pMO-LS20	Preference-based Multi-objective EA with PSO applied at every 20 generations
9	pMO-LS100	Preference-based Multi-objective EA with PSO applied at every 100 generations

Table 4.5: Description of the R_f values for the various problems (optimal F_3 highlighted in parentheses).

Problem	PORT1	PORT2	PORT3	PORT4	PORT5
25% R_f	0.0034	0.0030	0.0026	0.0028	0.0010
	(3.2402)	(11.0774)	(8.2546)	(9.1543)	(4.0154)
50% R_f	0.0068	0.0059	0.0053	0.0056	0.0020
	(0.9591)	(4.0934)	(2.5864)	(2.6803)	(2.3330)
75% R_f	0.0102	0.0089	0.0079	0.0083	0.0030
	(0.1315)	(0.3799)	0.1835)	(0.3351)	(0.8283)

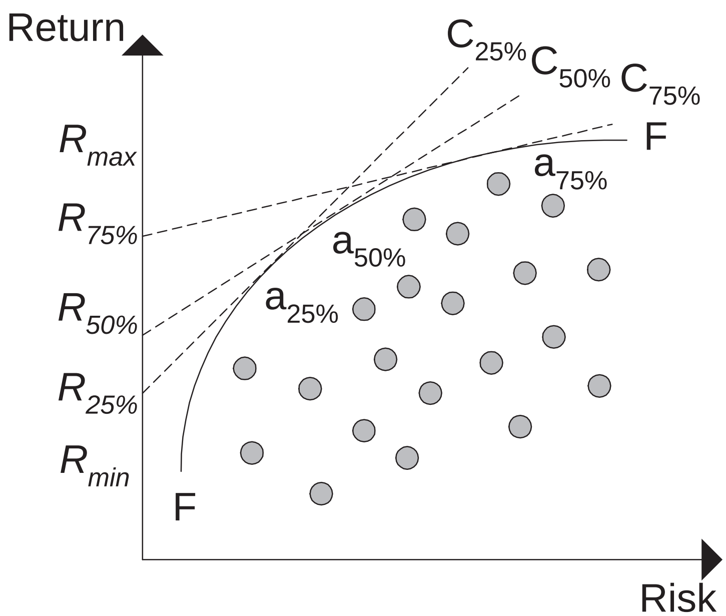
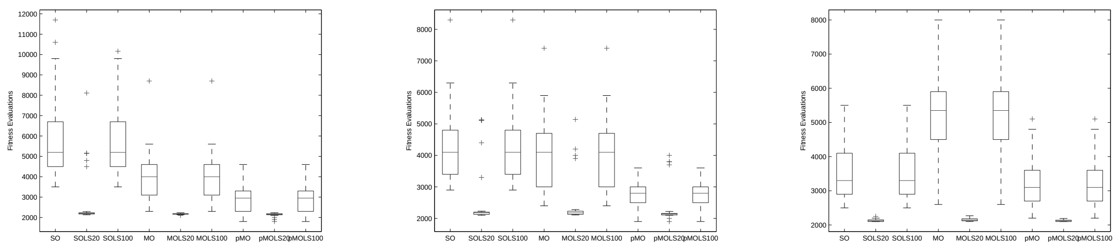


Figure 4.11: Illustration of different risk-free returns considered and the corresponding optimal solution.

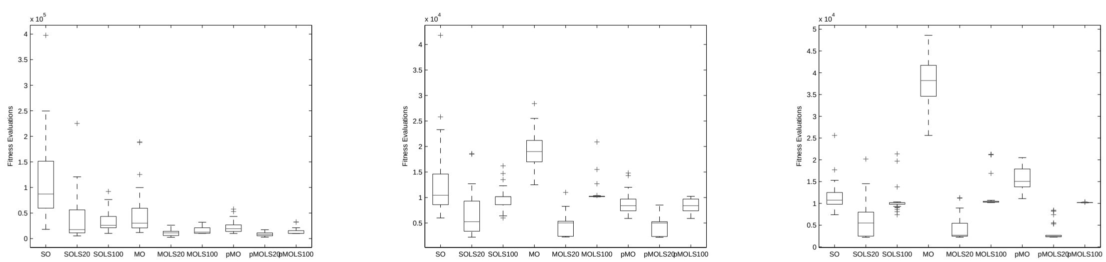
performance difference between SO and MO. For example, the significant algorithmic improvement in PORT2 for 75%-R_f corresponded to the huge performance difference between SO and MO, while the contrary was observed for 25%-R_f. The performance difference actually reflected the difficulty in locating the efficient portfolio on the efficient frontier and hence the level of improvement attainable when local optimizer is implemented. As such, application of pMO will be more justified in the case of when there is huge performance difference between SO and MO. Nevertheless, overall, pMO was able to converge faster to the efficient portfolio as compared to SO, yet at the same time, offer more alternatives in the vicinity of the preferred region.

Generally, the application of local search improved the convergence time of the algorithms. Nevertheless, this is dependent on G_{local}, where more significant improvements are generated when the local search is triggered more frequently, in accordance to the earlier observations. This was especially prominent in simpler problems (lesser number of assets) like PORT1, where the algorithm had converged before the local search could apply any positive effects on it. Overall, the application of both preference selection criteria and local search resulted in the fastest convergence to the efficient



(a) (b) (c)

Figure 4.12: Fitness evaluations required to reach within 5% of the optimal fitness for the various algorithms in PORT1 with (a) R_f = 0.0034 (b) R_f =0.0068 (c) R_f =0.0102.



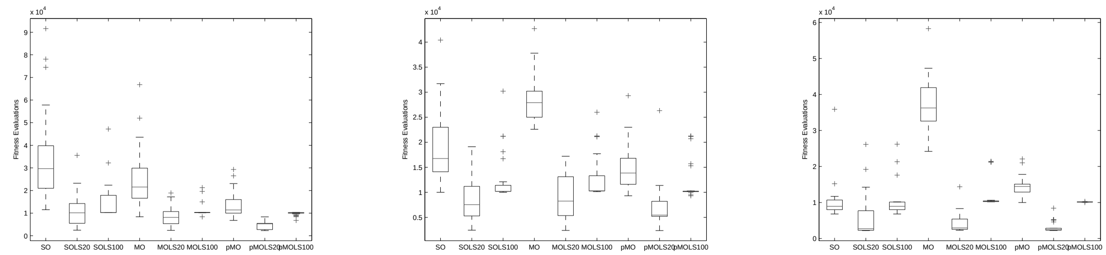
(a) (b) (c)

Figure 4.13: Fitness evaluations required to reach within 5% of the optimal fitness for the various algorithms in PORT2 with (a) R_f = 0.0030 (b) R_f = 0.0059 (c) R_f = 0.0089.

portfolio.

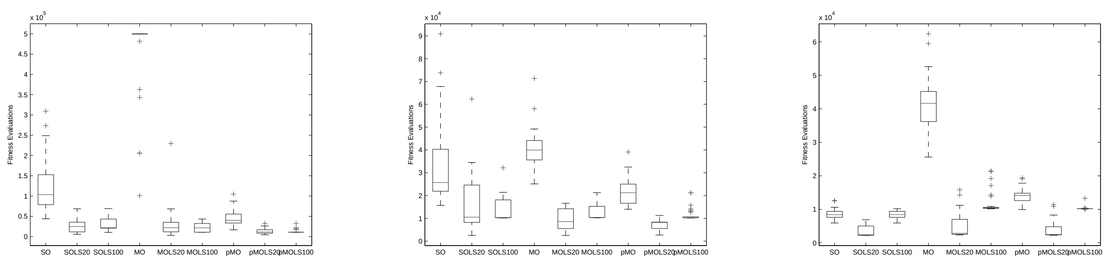
Interestingly, performance improvements were generally more significant for higher value of R_f. This could be due to the higher return and lower risk associated with higher R_f, which correspond to portfolio of smaller assets. The significant performance improvements corresponded to the observations in section 3.4.5 for global binary vector with higher P_{zero}.

EF_{Known} attained under the various algorithmic configurations are plotted in Figure 4.17. Clearly, a generic application of EA will have difficulty converging to the efficient portfolio in this case. However, with the implementation of local search or preference selection criteria, more solutions were found near the preferred region with the latter being better. Nevertheless, with the implementation of both local search and preference selection criteria, better solutions were attained. This



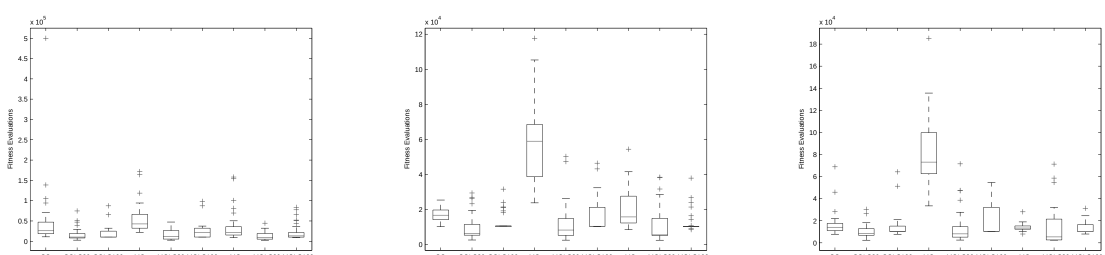
(a) (b) (c)

Figure 4.14: Fitness evaluations required to reach within 5% of the optimal fitness for the various algorithms in PORT3 with (a) R_f =0.0026 (b) R_f=0.0053 (c) R_f =0.0079.



(a) (b) (c)

Figure 4.15: Fitness evaluations required to reach within 5% of the optimal fitness for the various algorithms in PORT4 with (a) R_f =0.0028 (b) R_f =0.0056 (c) R_f =0.0083.



(a) (b) (c)

Figure 4.16: Fitness evaluations required to reach within 5% of the optimal fitness for the various algorithms in PORT5 with (a) R_f =0.0010 (b) R_f =0.0020 (c) R_f =0.0030.

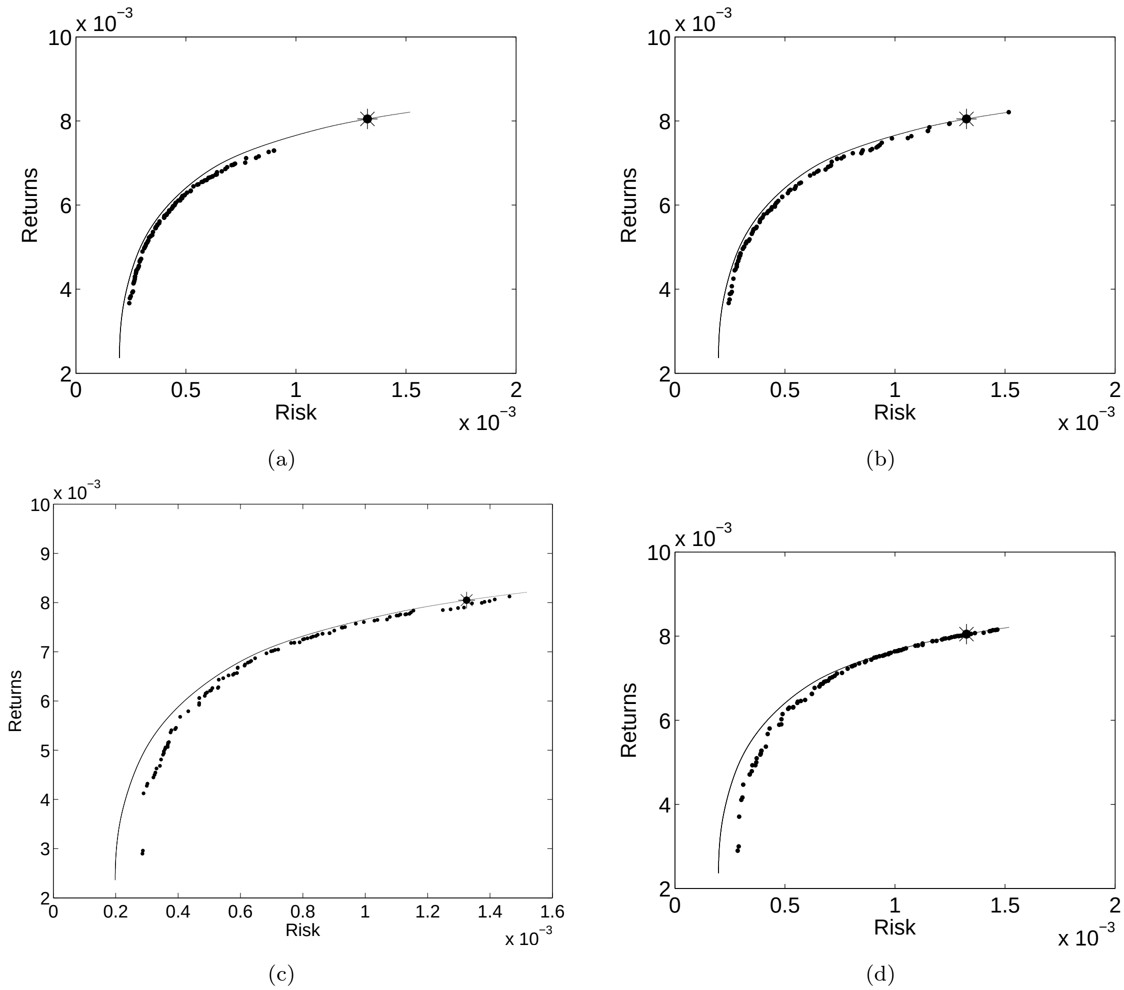


Figure 4.17: EF_{Known} attained by (a) MO, (b) MOLS20, (c) pMO and (d) pMOLS20 for PORT3 with $R_f = 0.0079$ within 10,000 fitness evaluations. The corresponding efficient portfolio and the efficient frontier are denoted by the star and dotted-line respectively.

is shown clearer in Figure 4.18, which presents a close-up illustration of EF_{Known} in the preferred region.

4.6 Summary

This chapter casts the Markowitz model into the multi-objective optimization domain and studied the application of the evolutionary platform discussed in Chapter 3. Generally, the ordered-based

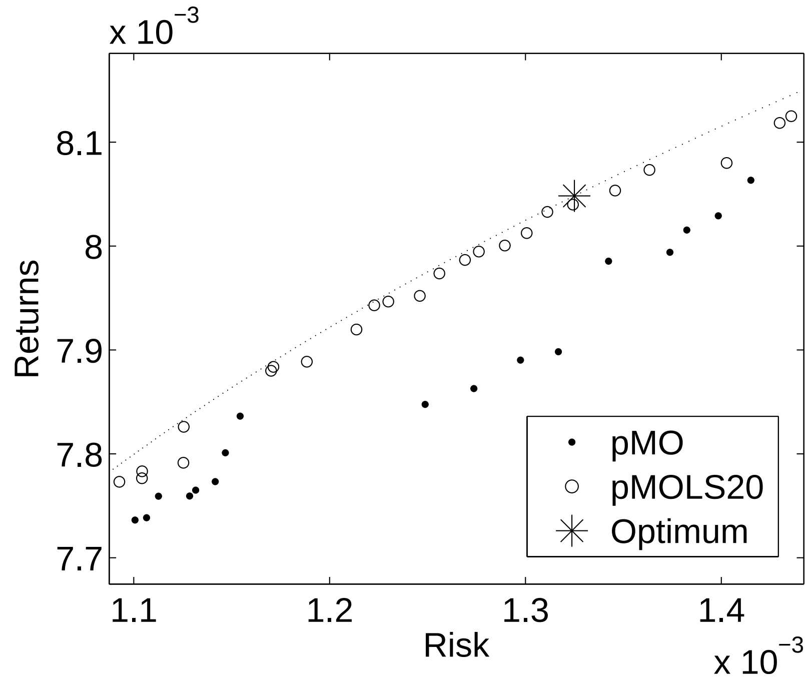


Figure 4.18: Close-up illustration of EF_{Known} attained by pMO and pMOLS20 in the preferred region. The corresponding efficient portfolio and the efficient frontier are denoted by the star and dotted-line respectively.

representation was able to find reasonable approximation for the efficient frontier in the various data sets. Lastly, the incorporation of preference-based techniques into the proposed evolutionary platform enhances its capability as a decision support system for portfolio managers in real-world implementation. Nevertheless, it is necessary to extend the evolutionary optimization model to handle other realistic constraints like round-lot constraints and transaction costs and evaluate its viability and practicality on more comprehensive test problems. This will be addressed in the next chapter.

Chapter 5

Handling Realistic Constraints in Portfolio Optimization

5.1 Introduction

Although Markowitzs mean-variance model of portfolio selection represents one of the best known models in finance, this simplistic model failed to describe the complexities of the real world in an adequate fashion, as portfolio managers often face various constraints in the real world of investment management arising from pre-specified investment mandates, business/industrial regulations and other practical issues [221]. Many of these constraints have been deliberately left out in academic studies, mainly to accommodate the limitations of classical optimization techniques.

One important aspect in investment portfolio management is that most of the risk diversification in a portfolio can be achieved with a small, yet well chosen, set of assets [74]. Empirically, systematic risk, i.e. the proportion of risk dependent only on the market becomes negligible when the number of assets in a portfolio exceeds approximately 20-25, though there have been recent evidences that this threshold might have increased to around 50 in recent years [32]. As such, asset allocation is often preceded by the question of whether to include the asset in the first place. The inclusion of cardinality constraints (i.e. explicit restrictions on the number of different assets in the portfolio) in the Markowitz mean-variance model will result in a mixed integer nonlinear optimization problem, which is typically unsolvable under classical optimization tools. While recent developed optimization techniques allowed such constraints to be considered, in-depth analysis that examined how these constraints affect the search progress and the efficient frontier attainable are sorely lacking.

Apart from the diminishing risk premiums, large portfolio size will typically involve higher management and maintenance costs, for example transaction costs, which is an inevitable factor in the construction and management of investment portfolios. While related literature works have modeled transaction costs as a fixed charge and/or a proportional element that is directly linked to the traded volume, transaction costs in real-world market practices often comprised of multi-tiered structures that are significantly more complicated to model. In fact, the precise treatment of transactions costs in portfolio optimization will lead to a non-convex problem landscape, in which numerical methods will typically fail. An intuitive extension after transaction costs will be the consideration of round-lot constraints (i.e. assets in the portfolio to be in exact multiples of the trading lots), which will further complicate the underlying optimization problem.

This motivation of this chapter is to incorporate the above-highlighted constraints into the Markowitz mean-variance model and evaluate the capability of the multi-objective evolutionary platform in handling these constraints, hence providing a viable platform to study the nature of the portfolio optimization problem in greater depths. Most importantly, the multi-objective formulation of the portfolio optimization problem allows the study of the entire efficient frontier attainable under the various constraints, which is otherwise not possible under conventional single-objective approaches. Specifically, two sets of constraints will be investigated, namely cardinality constraint with buy-in thresholds, followed by round-lot constraint with transaction cost. Their corresponding sections will discuss their incorporation into the problem model, the associated constraint handling techniques, followed by empirical analysis.

5.2 Review of Realistic Constraints in Portfolio Optimization

In the real world of investment management, portfolio managers often face a number of realistic constraints arising from pre-specified investment mandates, business/industrial regulations and other practical issues [221]. Examples of such constraints include (not exhaustively) floor and ceiling constraint, cardinality constraint, round-lot constraint, transaction cost inclusion, turnover constraint, trading constraint [163].

5.2.1 Floor and Ceiling Constraints

The floor and ceiling constraint, otherwise known as buy-in thresholds, specify the lowest and highest limits on the proportion of each asset that can be held in a single portfolio. 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 any one portfolio constituent as part of institutional diversification policy [57]. This constraint is formulated as such,

$$a_i \leq w_i \leq b_i, 0 \leq a_i \leq b_i \leq 1 \tag{5.1}$$

where a_i and b_i denotes respectively the minimum and maximum weights that can be held for asset i $(i = 1, 2.., N)$ in the portfolio. While floor constraint has been actively studied in related literature [7, 72, 206, 221, 225, 226, 227, 228], the general floor and ceiling constraint has been less explored.

5.2.2 Cardinality Constraint

Investors and fund managers often limit the number of assets in their portfolio for ease in management and monitoring and/or to avoid excessive operational and maintenance costs. On the other hand, a limit on the minimum number of assets in the portfolio will be required to capture diversification benefits. Cardinality constraints denote the limits on the maximum and minimum numbers of assets that a portfolio can hold and they can be expressed as follows:

$$Car_l \leq \sum_{i=1}^{N} z_i \leq Car_u, 0 \leq Car_l \leq Car_u \leq N \tag{5.2}$$

where $z_i = 1$ if $w_i > 0$ and $z_i = 0$, otherwise and Car_l and Car_u respectively denote the lower and upper cardinal limit. This constraint has been simplified in several related works, where either the inequality restriction in 5.2 is replaced by an equality restriction instead i.e. portfolios are restricted to a pre-specified cardinal size [7, 37, 72, 163, 225, 226, 227, 228] or only the maximum cardinality constraint is considered [221, 206, 164].

5.2.3 Round-lot Constraint

Round-lot constraint requires the number of each asset in the portfolio to be in exact multiples of the trading lots [152, 221]. The round-lot constraint can be expressed as

$$k_i \mod l_i = 0 \tag{5.3}$$

where k_i denotes the number of units purchased for asset i and l_i represents its corresponding lot size. Typically, the inclusion of round-lot constraint will require a relaxation of the budget constraint (4.3) as the total capital will not be of exact multiples of the minimum lot prices for the various assets.

5.2.4 Turnover Constraint

Turnover constraints are trading limits defined by practitioners to safeguard against excessive transaction cost slippages [212]. Essentially, they are upper variation bounds for asset holding in the portfolio between trading periods. The implicit assumption behind this indirect control of transaction costs is that if proportional transaction cost is considered and they are equal across assets, controlling turnover is analogous to controlling transaction cost. Turnover constraints can be expressed as such

$$max(w_i - w_i^0, 0) \leq \overline{B_i} \tag{5.4}$$

$$max(w_i^0 - w_i, 0) \leq \overline{S_i} \tag{5.5}$$

where w_i^0 represents existing holding proportion of asset i prior to the portfolio construction, while $\overline{B_i}$ and $\overline{S_i}$ denote respectively the maximum purchase and sales of asset i between trading periods. Generally, $\overline{B_i}$ and $\overline{S_i}$ will vary in different period in accordance to the market conditions.

5.2.5 Trading Constraint

Contrary to turnover constraints, trading constraints impose minimum limits on buying and selling of tiny quantities of assets due to practical reasons, for example, high fixed costs linked to transaction

[57]. As such, higher transacted volumes is required to distribute the overheads. Trading constraints can be expressed as such

$$w_i = 0 \text{ or } w_i \geq w_i^0 + \underline{B_i} \tag{5.6}$$

$$w_i = 0 \text{ or } w_i \leq w_i^0 - \underline{S_i} \tag{5.7}$$

Where w_i^0 represents the holding proportion of asset i prior to the portfolio construction, while $\underline{B_i}$ and $\underline{S_i}$ denotes respectively the minimum purchase and sales of asset i between trading periods. Generally, the incorporation of transaction costs into the optimization problem will indirectly control the turnover and trading constraints.

5.2.6 Transaction Costs

Transaction costs associated with the purchase and sales of assets are inevitable in real-world practices. Typically, these costs could be variable and/or proportional to the traded volume. In other cases, a lower limit (i.e. minimum fee per transaction) might be imposed and/or they might come with a fixed component (i.e. fixed fee per transaction). Mathematically, the various transaction cost function can be expressed as such,

$$TC_t = f(v) = \begin{cases} \eta & \text{,fixed cost only} \\ \lambda v & \text{,proportional cost only} \\ max\,(\lambda v, \eta) & \text{,proportional cost with lower limit} \\ \eta + \lambda v & \text{,proportional plus fixed cost} \end{cases} \tag{5.8}$$

for some constant η, λ, where $0 < \eta < 1$ and $\lambda > 0$ and for some traded value v. Although the various cost structures have been considered extensively in related academic works, they do not correspond exactly to actual market practices, where transaction costs often comprised of a multi-tiered cost structure with different cost functions applied within different ranges of the traded value. Such formulation will lead to discontinuities and/non-concavity in the set of feasible portfolio selection, where traditional approaches like quadratic or linear programming will fail. As such, heuristic and evolutionary approaches were considered [57, 161] instead, which are capable of handling such problem landscapes.

5.3 Handling Cardinality Constraint with Buy-in Threshold

The cardinality constraints and the buy-in threshold will be considered here. The section will begin with a brief discussion on the corresponding constraint handling techniques, before presenting the empirical analysis on the effects of the efficient frontier attainable under these constraints.

5.3.1 Constraint Handling Technique for Buy-In Threshold

This constraint requires asset weight values to be within a specific range. As such, the conventional strategy of normalizing the total weight to one so as to meet the budget constraint is no longer applicable here, since the normalized weights might not be within the limits. Related works focus only on floor constraint and the conventional approach is to arbitrarily add the minimum weight to any infeasible assets. A simple technique is proposed here to handle the general floor and ceiling constraints, which involves modifying the fitness evaluation operation whilst maintaining the same representation and variation operation and other evolutionary operators.

The proposed order-based representation allows direct control on the manner in which the assets are introduced into the portfolio and any infeasibility can be immediately repaired. As such, the floor and ceiling constraints are regarded as hard constraints here. The modified fitness evaluation will still initialize an empty portfolio. Subsequently, assets will be added iteratively, with their corresponding weights adjusted to the floor and ceiling constraint as shown in (5.9).

$$w_i^a = a_i + (b_i - a_i) \times w_i \tag{5.9}$$

Assets will be added to the portfolio until the total weight of the portfolio exceeds one. At this stage, case-dependent correction techniques will be applied to ensure the feasibility of the final portfolio constructed. A total of 3 different cases have been identified:

1. After removing the last added asset, the remaining weight is between the floor and ceiling limits: In this case, the weight of the last added asset can simply be reassigned so that its adjusted weight is equivalent to the remainder needed to attain a total weight of one.

2. After removing the last added asset, the remaining weight is less than the floor limits: This case can be further subdivided into two different scenarios.

   (a) After removing the second last added asset, the remaining weight is between the floor and ceiling limits: This will mean that its weight can simply be reassigned so that the adjusted weight is equivalent to the remainder that it needed to attain a weight counter of one. The portfolio will contain all the assets considered so far and the adjusted asset.

   (b) After removing the second last added asset, the remaining weight is outside the floor and ceiling limits: for this case, all the weight vectors will simply be readjusted by either increasing or decreasing them by a predefined percentage.

This modified fitness evaluation will ensure that all the solutions generated during the evolutionary search progress will always be feasible with respect to this constraint.

5.3.2 Constraint Handling Technique for Cardinality Constraint

Related works in literature have considered cardinality constraint as a hard constraint and generalized the inequality restriction by an equality constraint. As such, a fixed number of assets are arbitrarily selected based on the fixed cardinal value before the weights are normalized to satisfy the other constraints. Similar techniques can be employed to satisfy the maximal cardinality constraints [164, 206, 221] by setting the weights of excess assets to be zero. However, such techniques might have difficulties dealing with ceiling constraints as the excess weights cannot be arbitrarily assigned to other assets.

This chapter considers the general cardinality constraint as a soft constraint instead. Repair operation, as described in Figure 5.1, is used to correct the feasibility of the chromosome. Specifically, the various values in the weight vector will be increased/decreased when its associated portfolio size is too high/low, so that fewer/more assets will be required in the re-evaluation. This simple procedure will help to adjust the portfolio size of infeasible chromosomes back to the feasible range.

Due to the presence of infeasible solutions in the evolving population, the selection operation will have to factor this into consideration also. Basically, the feasibility of the portfolio will take priority over the optimality of the solutions. This is applicable for both the parent selection and the survivor selection.

```
IF number of asset > maximum cardinal
    Increase all weights by k %
else IF number of asset < minimum cardinal
    Decrease all weights by k %
end IF
```

Figure 5.1: Pseudo code of the repair operation for cardinality infeasibility.

5.3.3 Simulation Result & Discussion

Simulation results in Section 4.4.3 demonstrated the capability of the proposed order-based representation in generating better approximation of the efficient frontier as compared to other representations. This section will extend the evolutionary platform to the constrained portfolio optimization model and evaluate its constraint handling ability with respect to the floor and ceiling constraint and cardinality constraint. Particularly, this study will be restricted to PORT3 and the generational stopping criteria will be extended to 1000 to ensure algorithmic convergence.

Before examining simulation results, it will be instructive to analyze how portfolio size changes along the efficient frontier in general. Figure 5.2 plots the portfolio risk against its size for all the solutions obtained by OR3 for PORT3 in 30 runs. Clearly, smaller portfolio sizes were associated with higher risk bands while larger portfolio sizes possessed smaller risk due to diversification. As such, the imposition of floor and ceiling constraint and cardinality constraint, which limit the portfolio sizes, will influence the level of return and risk attainable, thus restricting the optimal solutions to certain regions of the original efficient frontier.

Particularly in the context of floor and ceiling constraint, the former will force a minimal exposure to those lower-returning assets, while the latter will prevent the high optimal level of exposure to high returns assets, again forcing an exposure to lower-returning assets. This will ultimately reduce the overall portfolios return, resulting in suboptimal portfolio. To verify this hypothesis, two sets of floor and ceiling constraints: $\{1\%, 2\%\}$ and $\{10\%, 11\%\}$ were considered and the constrained EF_{known} are shown in Figure 5.3. Clearly, with this constraint, it was not possible to approximate the entire efficient frontier and EF_{known} attained were limited to the low risk-return region.

To further investigate the effects of floor and ceiling constraint on portfolio sizes, different values of the constraint were considered and the average portfolio sizes obtained under the various instances

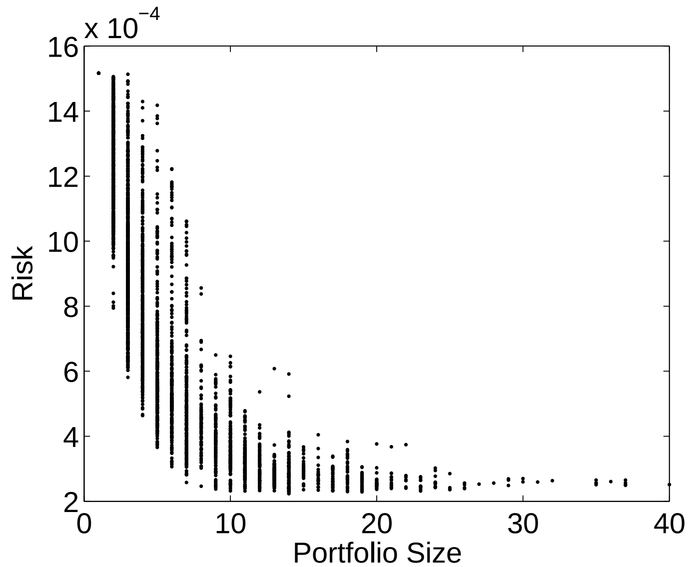


Figure 5.2: Plot of risk against portfolio size obtained by OR3 in PORT3.

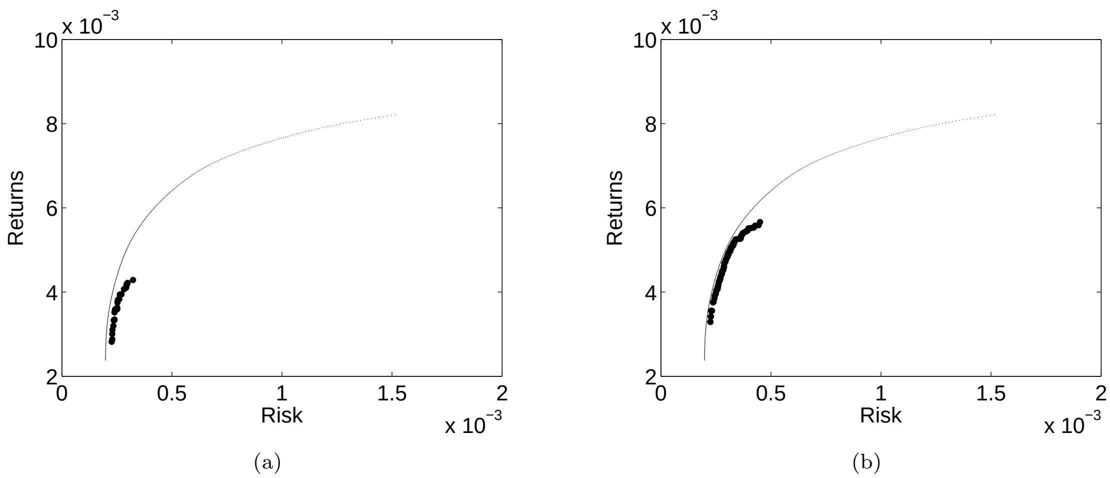


Figure 5.3: Constrained EF_{known} attained for PORT3 with floor and ceiling constraint of (a) {1%, 2%} and (b) {10%, 11%}, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

is shown in Figure 5.4. The dark region denotes the infeasible case where the floor constraint was higher then the ceiling constraint. With the floor and ceiling constraint, the average portfolio size generally increased as compared to 7.02 in the unconstrained case. By using a higher ceiling constraint, larger weights values were possible, resulting in the reduction of the portfolio size. Similarly, increasing the floor constraint will have the same effect as larger weights values were required. Comparing the two set of constraint considered, {1%, 2%} attained a larger portfolio size, resulting in the attainable Pareto front to be situated near the low risk and return region in Figure 5.3(a) due to excessive diversification. On the other hand, increasing the constraints values to {10%, 11%} allowed higher riskreturn portfolio to be attained and stretched the attainable EF_{known} upwards as shown in Figure 5.3(b).

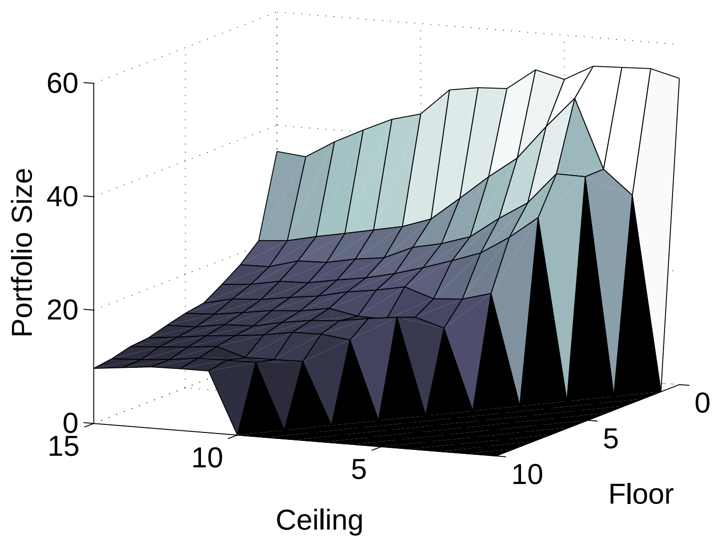


Figure 5.4: Average portfolio size obtained for various values of floor and ceiling constraint.

Contrary to the floor and ceiling constraints, cardinality constraint influences the portfolio size directly. As such, the cardinality-constrained efficient frontier might be discontinuous, as certain portfolios will not be available for the rational investor [37]. Figure 5.5 shows the effects of adopting a fixed cardinality constraint and the discontinuity phenomenon is clearly evident here where the tight cardinality constraint confined the constrained EF_{known} to the high risk region as efficient risk diversification is ruled out.

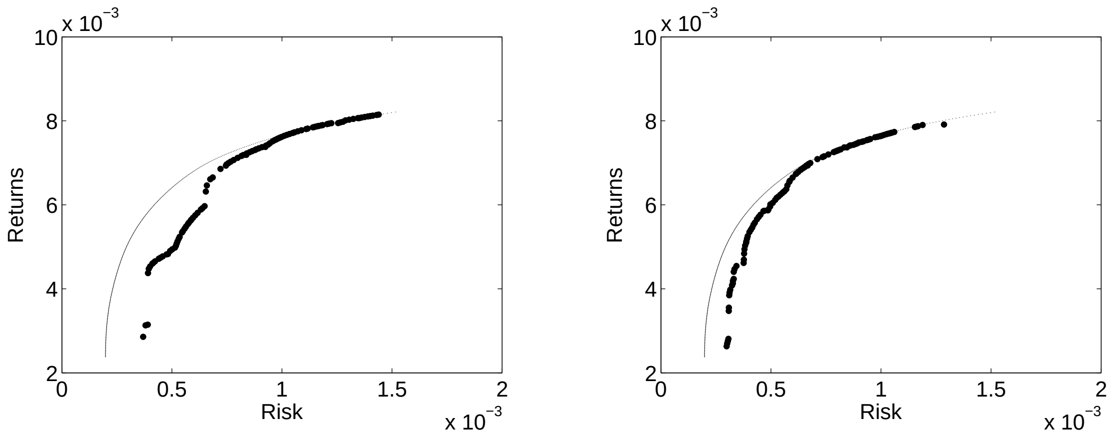


Figure 5.5: Constrained EF_{known} attained for PORT2 with cardinality constraints (a) {2, 2} (b) {3, 3}, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

However, as the cardinality limits were relaxed, the constrained EF_{known} became more continuous as illustrated in Figure 5.6 where the constraints were relaxed to {2,3} and {1,4} respectively. However, the low risk-return regions were not very well defined since large portfolio sizes were not possible under these constraints. Nevertheless, it should be highlighted that the actual effects on the constrained frontier ultimately depend on the extent of relaxation in the cardinality constraints. Similar observations were witnessed in the case for higher values of cardinality constraints, where the only difference was that the constrained EF_{known} were now confined to the low-risk region.

To further evaluate the generality of the constraint handling technique, both floor and ceiling constraint and cardinality constraint was considered together. Previous result have shown that for a floor and ceiling constraint of {1%, 12%}, the portfolio size ranged from 15 to 35 with a mean value of 23. Adopting this value of floor and ceiling constraint, different cardinality constraints were considered and the constrained EF_{known} were compared to that obtained without the cardinality constraint. Generally, the proposed constraint handling technique was able to attain an EF_{known} that satisfied both the constraints. Different level of cardinality constraints restricted the constrained EF_{known} to different risk-return region i.e. the cardinality constraint {15, 20} which was below the mean portfolio size corresponded to the higher risk-return region while the higher cardinality constraint corresponded to the lower risk-return region. Figure 5.8(c) shows that if the cardinality

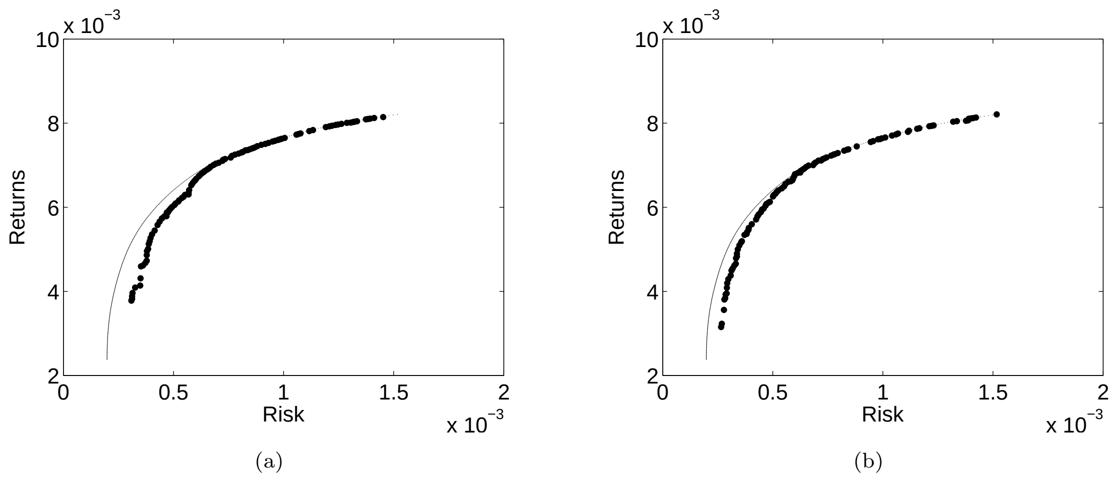


Figure 5.6: Constrained EF_{known} attained for PORT3 with cardinality constraints (a) {2, 3} (b) {1, 4}, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

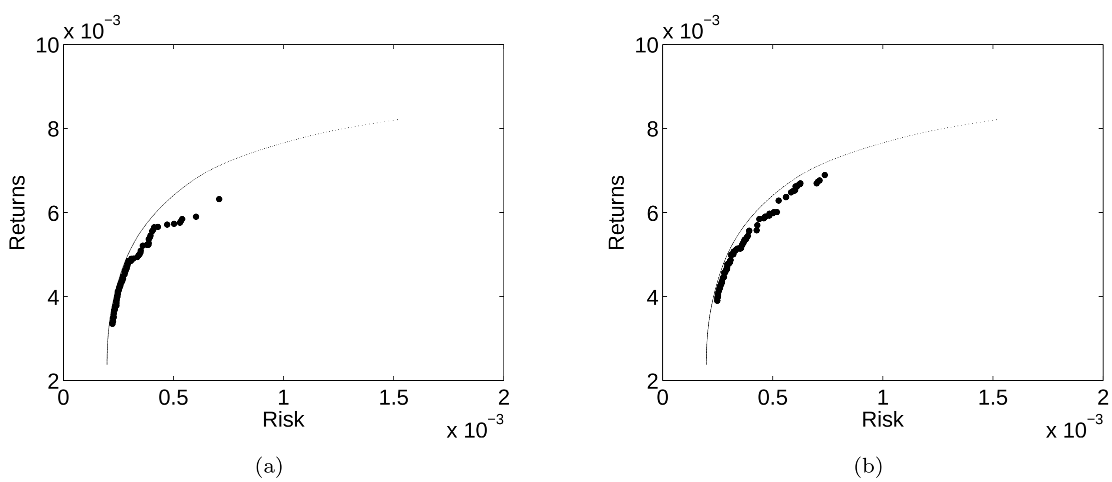
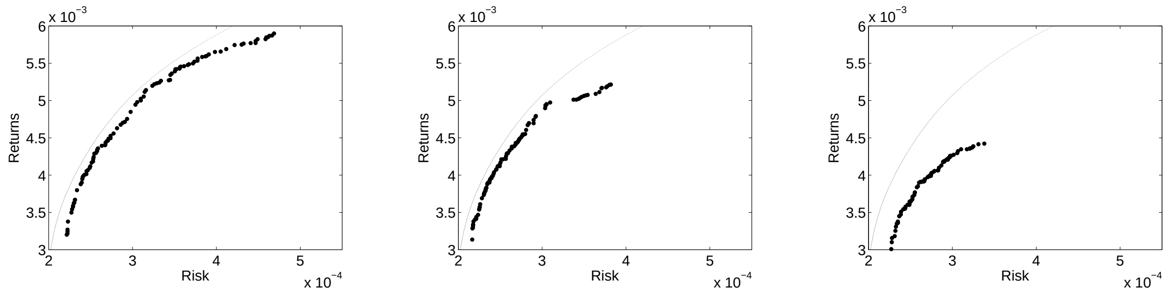


Figure 5.7: Constrained EF_{known} attained for PORT3 with cardinality constraints (a) {35, 35} (b) {32, 38}, with the corresponding unconstrained EF_{True} denoted by the dotted-line.



(a) (b) (c)

Figure 5.8: Constrained EF_{known} attained for PORT3 with combined floor and ceiling constraints and cardinality constraints respectively at {1%, 12%} and (a) {15, 20}, (b) {25, 30} and (c) {50, 55}, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

constraint was fixed outside the optimal portfolio size range, it will result in a suboptimal Pareto front. It should be highlighted that if these constraints are too rigid, there might be a possibility that there will not be any feasible portfolio. Take for example a maximum cardinality of 3 and a ceiling constraint of 0.1, the minimum portfolio size based on the latter i.e. 10 could not possibly satisfy the cardinality constraint under any circumstances.

5.4 Handling Round-Lot Constraint with Transaction Costs

Most portfolio selection models in related literature works assumed the perfect fractionability of security values, where each asset can be conveniently represented by a real variable. However in market practices, securities are instead negotiated as an integer multiple of a minimum transaction lot. Essentially, lot size represents the standardized quantity of a financial instrument as specified by the associated regulatory body and it denotes the minimum quantity in which the security may be traded. For example, one lot typically denotes 1,000 units of share in Singapore, though the lot sizes varies for different asset classes and markets. Without any loss in generality, lot size of 1,000 will be the standard convention for the rest of this chapter.

More often than not, related works in literature often denotes transaction costs as a fixed fee component [36, 70, 183, 213] and/or a variable fee that is proportional to the traded amount [2, 59, 69, 219]. However, this does not correspond to actual market practices, where transaction costs

Table 5.1: Transaction fee structure for the four major brokers in Singapore. Information was extracted from their corresponding websites as of 14/04/2008.

Traded Value	POEMS (Non-Advisory)		DMG & Partners Securities Pte Ltd		DBS Vickers		UOB
	Online	Offline/ Phone	Online	Offline/ Phone	Online	Offline/ Phone	Online
Minimum	25	40	25	40	25	40	25
less than 50k	0.280	0.375	0.275	0.500	0.280	0.375	0.275
50k to 100k	0.220	0.300	0.220	0.400	0.220	0.300	0.275
more than 100k	0.180	0.225	0.180	0.250	0.180	0.225	0.275

instead comprised of a multi-tiered cost structure where different cost functions applied within certain ranges of the traded volume as shown in Table 5.1, which provides a breakdown of the transaction fee structures for the four major brokers in Singapore. Firstly, a fixed minimum cost is required if the traded value falls below a critical value. Secondly, the proportional constant will decrease for higher traded values to encourage higher trading activity for their clients. Lastly, depending on the trading medium, there will be different tiered cost structure.

The transaction cost structure of POEMS can be mathematically formulated as a set of piecewise affine functions [143]. Without any loss in generality, this will be the formulation considered in this chapter also.

$$TC(v_i) = \begin{cases} 0 & v_i \leq 0 \\ 25 & 0 < v_i \leq 25 \\ 0.0028v_i & 25 < v_i \leq 50000 \\ 0.0022v_i & 50000 < v_i \leq 100000 \\ 0.0018v_i & 100000 < v_i \end{cases} \tag{5.10}$$

where v_i denotes the transacted value of asset i. The impact of transaction cost will be a function of the investors initial capital, as it will chalk up a huge proportion of the portfolio value if the capital is excessively small. Consequently, the calculation of the expected returns has to be re-adjusted to compensate for transaction costs also.

5.4.1 Problem Formulation

The mean variance model considered earlier in chapter 4 will be extended here to include transaction costs and lot constraints, mathematically formulated in (5.11) to (5.14) below. The variables notations are listed in Table 5.2.

$$\min f_1 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} \tag{5.11}$$

$$\min f_2 = \frac{\sum_{i=1}^{N} p_i k_i \mu_i - \sum_{i=1}^{N} TC(v_i)}{C} \tag{5.12}$$

subjected to

$$k_i \mod l_i = 0 \tag{5.13}$$

$$\sum_{i=1}^{N} p_i k_i \mu_i + \sum_{i=1}^{N} TC(v_i) \leq C_{max} \tag{5.14}$$

Table 5.2: Variable notations for the portfolio optimization problem.

Variable Notation	Description
N	Total number of distinct assets available
C_{max}	Maximum capital available to the investor for investment
C	Capital utilized in the portfolio build-up (excluding transaction costs)
p_i	Market price of asset i
k_i	Number of asset i included in the portfolio
l_i	trading lots of asset i
w_i	Composition of asset i in the portfolio as a proportion of the total available capital ($w_i = \frac{p_i k_i}{C}$)
σ_{ij}	covariance between assets i and j
μ_i	expected return of asset i
$TC(v_i)$	Transaction cost function (5.10), where $v_i = p_i \times k_i$

Essentially, the portfolio volatility remained unchanged (5.11), the price volatility of the individual assets are independent to transaction costs. On the other hand, the expected return will be reduced by the transaction costs incurred during the portfolio build-up, resulting in a downward shift in the efficient frontier. This will be investigated further later. (5.13) denotes the lot constraint. Consequently, the inclusion of (5.13) will require a relaxation of the budget constraint (5.14), where the aggregate portfolio value and the associated build up cost should be lower than the allocated budget. This formulation assumes that there are no prior holdings in the portfolio.

The expected returns, μ_i for each asset and covariance matrix were based on 291 weekly price data from 31 different stocks in the Hang Seng Index (Hong Kong) from the period of March 1992 to September 1997 [11]. μ_i for each asset was simply the compounded return (annualized) based on the historical data (5.15). Meanwhile, the covariance matrix was derived by considering the annualized historical covariance between the different assets (5.16). The factor $\sqrt{52}$ in (5.16) denotes the annualizing factor (assuming a 52 weeks base).

$$\overline{r_i} = \prod_{t=1}^{T}(1 + r_i)^{\frac{52}{T}} - 1 \tag{5.15}$$

$$\sigma = \sqrt{52} \times \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(\bar{(t)} - t_i)^2} \tag{5.16}$$

While more sophisticated methods to estimate the expected returns and covariance matrix are available in literature, they will not be considered here as this work focuses on the portfolio selection rather than the predicative capability of the model. Figure 5.9 depicts the efficient frontier and the expected returns and volatilities of the individual assets.

The proposed representation can be easily extended to handle the lot constraint, in a similar fashion to floor and ceiling constraints. Essentially, for every asset that is added into the portfolio, they will first be adjusted based on the floor and ceiling constraints. Following that, they will be rounded down to the largest weight available (5.17). Similar to techniques proposed by Skolpadungket et al. [2007], the remainder of the budget will be allocated to the assets in existing portfolio provided that the ceiling constraint is not satisfied and to the assets outside the portfolio if the floor constraint can be satisfied.

$$w_i'' = w_i' - \quad \bmod \frac{c_i}{C} \tag{5.17}$$

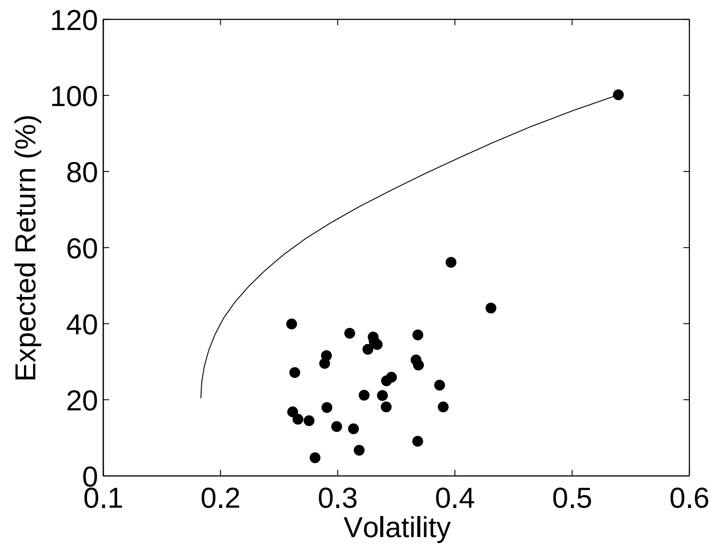


Figure 5.9: Volatility and Expected Return of considered stocks and the associated efficient frontier (line).

5.4.2 Simulation Result & Discussion

This section presents the simulation results to investigate the impact of lot constraints and transaction costs on the efficient frontier. The closing prices of the 31 assets in the dataset considered ranged from 8.25 to 410.04 with an average of 66.20. Based on an initial capital of 10k, a single lot of the cheapest stock will constitute 8.25% of the total portfolio value and a single lot of the most expensive stock could not be afforded at all. Technically, the effect of lot constraint is equivalent to setting floor and ceiling constraints to each and every stock in addition to a granularity level in which an investor can split his capital [163]. In this case, the conventional assumption of arbitrarily divisible assets does not apply here. On the contrary, large investors with more capital will enjoy finer granularity and less stringent floor and ceiling constraints. Clearly, the impact of the lot size constraint will be more significant for smaller investors as compared to larger investors.

To investigate this in greater details, various capitals i.e. {10k, 50k, 100k, 500k, 1000k, 5000k} were considered and their corresponding EF_{known} are plotted in Figure 5.10. Evidently, the capital size significantly influences the EF_{known} achievable, especially for $C \leq 10k$. Essentially, expensive securities will be ruled out with limited capital and replaced with cheaper alternatives, hence degrading the optimality of portfolio that required these securities. Also, the impact of lot constraints is more pronounced for portfolios at the lower risk level, which is of higher cardinal generally. This can be clearly seen in the case for C=10k, where the resultant EF_{known} is very similar to the unconstrained efficient frontier at high risk level, but is shifted downwards at the low risk level. Evidently, there seems to be a certain capital threshold where below which, there will be considerable impact on the efficient frontier.

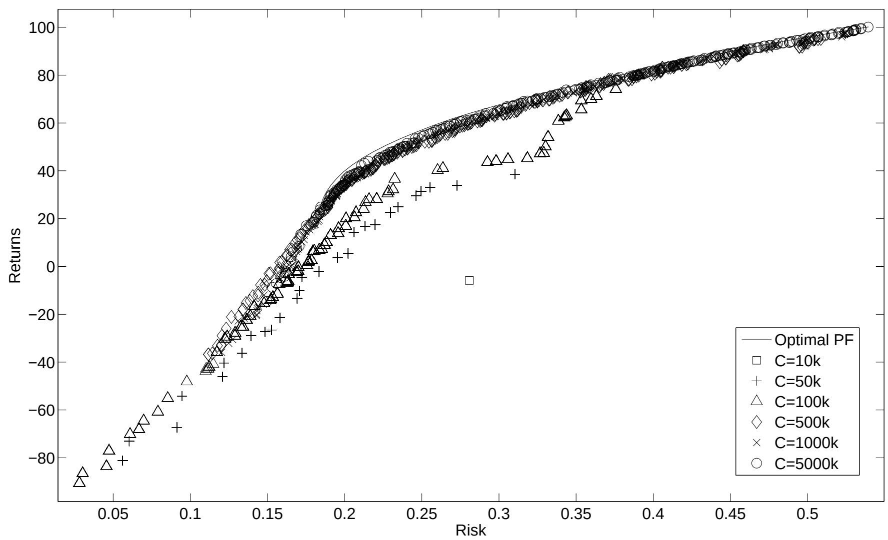


Figure 5.10: EF_{known} attained at the end of 30 algorithmic runs for lot size of 1000 at various level of C, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

A smaller lot size of 100 was considered under similar simulation settings and the resultant EF_{known} were plotted in Figure 5.11. Interestingly, EF_{known} for C=10k with lot size of 100 is identical to that when C=100k with lot size of 1000. This is expected as the two distinct problem settings essentially represents the same optimization problem i.e. reducing the lot size by a factor of 10 is equivalent to increasing the available capital by a factor of 10. Consequently, the reduced lot size relaxed the capital threshold where the efficient frontier will be considerably impacted. Also, the constrained EF_{known} remained highly similar to the unconstrained efficient frontier beyond that threshold.

To investigate the characteristics of the portfolios along the EF_{known} attained, Figure 5.12 plots

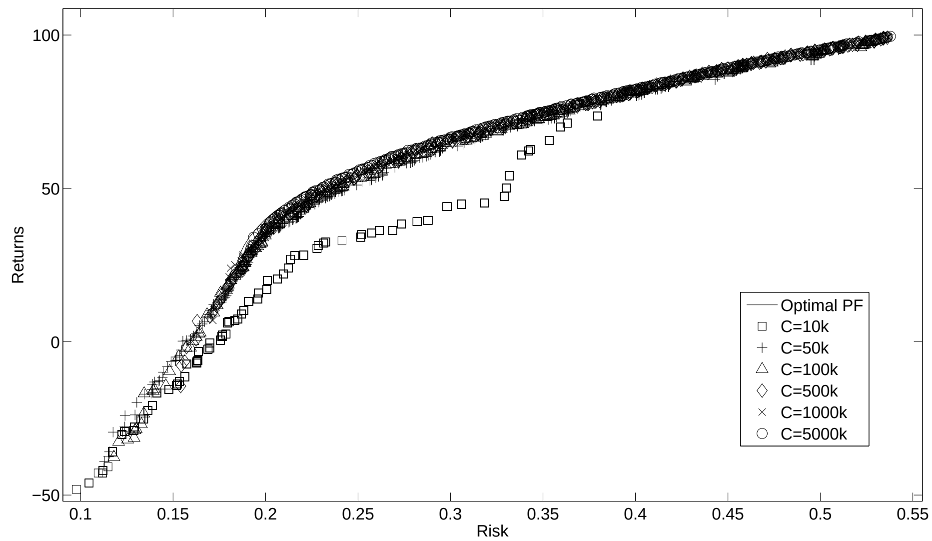
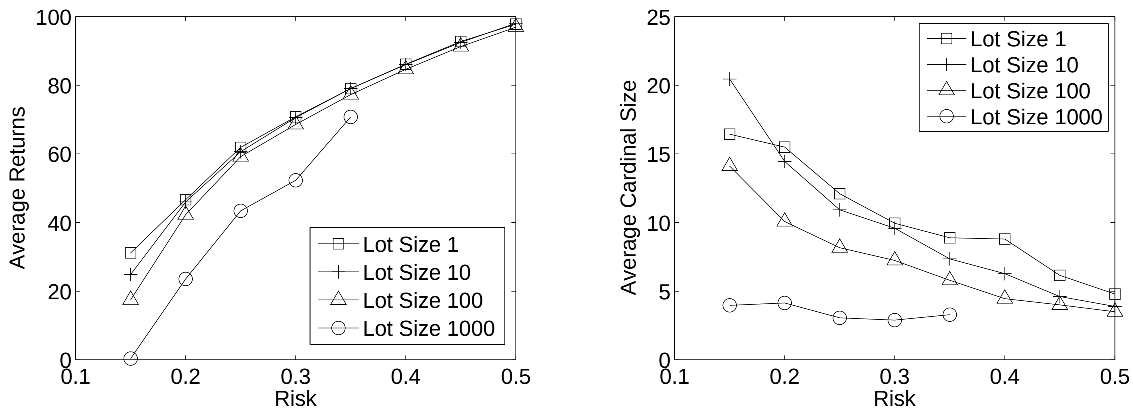


Figure 5.11: The efficient frontier attained at the end of 30 algorithmic runs for lot size of 100 at various level of capital, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

the average portfolio size and returns at risk intervals of 0.05 from 0.15 to 0.5. The capital is fixed at 100k here and a set of lot size was considered i.e. {1, 10, 100, 1000}. Generally, the cardinal size decreases as risk level increases, which is a key characteristic of risky portfolio discussed earlier. Also, larger lot sizes reduced the cardinal of the optimum portfolio at each risk interval, as investors are forced to assume larger positions, resulting in smaller portfolio size. This impact is especially significant for low risk level. Lastly, the similarity of EF_{known} for lot size 1 and 10 implies that the impact of the lot constraints is ultimately dependent on the lot size and the budget available.

Intuitively, the inclusion of transaction cost will deteriorate the efficient frontier further. As briefly discussed earlier, transaction cost will reduce the expected return of the portfolio, causing a downward shift in the efficient frontier. The impact is likely to be a function of C, similar to the lot constraint earlier. Intuitively, larger C will translate to a proportional cost structure as each security will be substantially invested. As such, the magnitude of the downward shift in the efficient frontier will be approximately equal to the lowest transaction rates i.e. 0.18% in this context. However, for smaller C, the efficient frontier will be shifted further downwards and due to transaction cost being structured as a set of piecewise affine functions, the shift will most likely to be non-linear.



(a) (b)

Figure 5.12: Average Cardinal Size (left) and Returns (right) at various levels of risk level (interval of 0.05) for different lot size at C=100k.

Transaction cost can constitute a significant proportion of the portfolio if the budget is extremely low.

Figure 5.13 plots the EF_{known} attained for different capital inclusive of transaction costs only. The set of capitals were considerably of small value so as to illustrate the effect of transaction cost. Clearly, lowering budget will result in a non-linear downward shift in EF_{known} as hypothesized and the impact is linear for high capital as shown in Figure 5.13, a close-up view on a particular portion in Figure 5.14.

Lot constraint and transaction cost are considered simultaneously and the resultant EF_{known} under different capital are plotted in Figure 5.15. The earlier results suggested that the detrimental effects of lot constraint stepped in earlier than transaction cost at higher C. Particularly at $C = 10K$, the inclusion of lot constraint result in significant changes in EF_{known} (Figure 5.10) as opposed to the transaction cost (Figure 5.13). As such, the EF_{known} in Figure 5.14 are seemingly similar to that in Figure 5.10. The insignificance of transaction cost is partly attributed to the single period instantiation of the portfolio optimization problem where only portfolio build-up cost is considered. Its significance will be more pronounced where they are cumulated in multi-period portfolio optimization, as the portfolio adjust its holding frequently in the investment horizon to adapt to the market conditions. This will be further investigated in the next chapter.

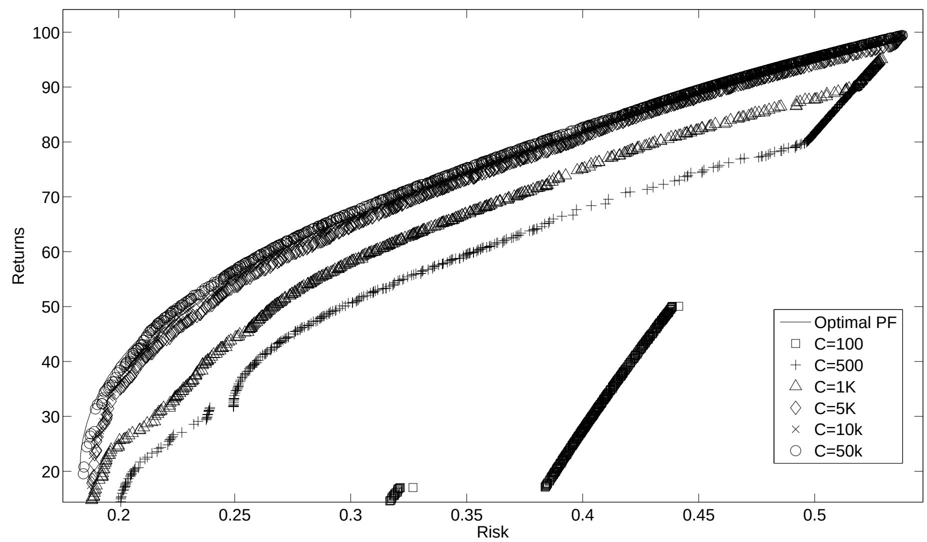


Figure 5.13: The efficient frontier attained at the end of 30 algorithmic runs for no lot with trans cost at various level of capital, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

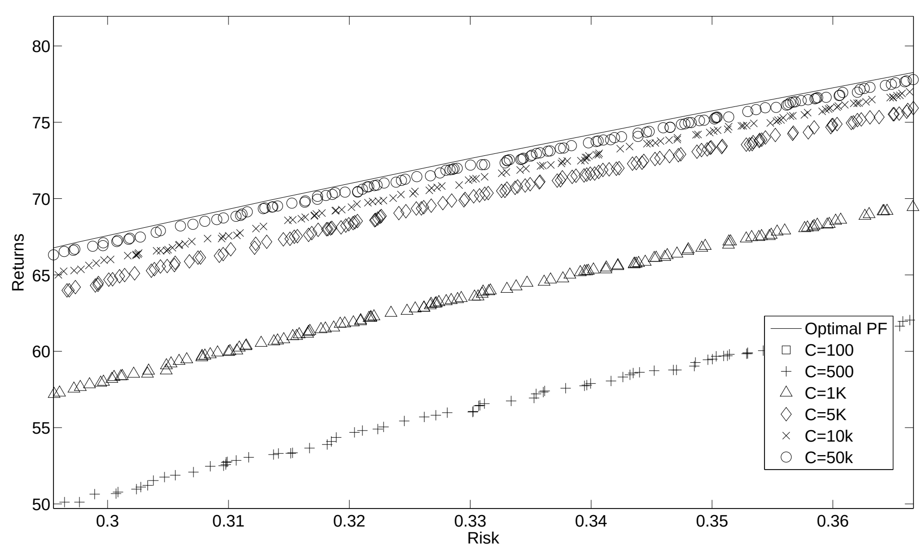


Figure 5.14: Close-up view of Figure 5.13.

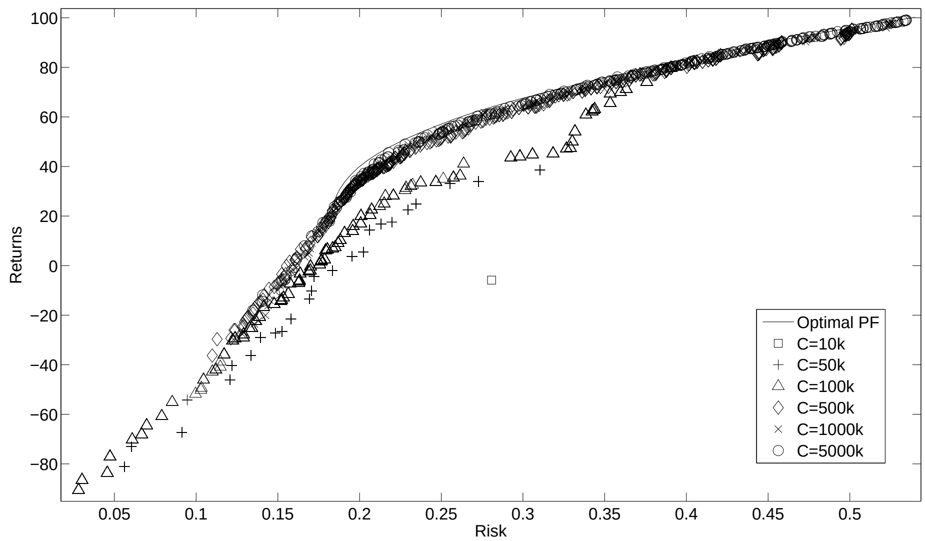


Figure 5.15: The efficient frontier attained at the end of 30 algorithmic runs for no lot with trans cost at various level of capital, with the corresponding unconstrained EF_{True} denoted by the dotted-line.

5.5 Summary

This chapter considered two set of constraints that are typically found in real-world investment portfolio management, namely cardinality constraint with buy-in thresholds and round-lot constraint with transaction cost. A discussion on the incorporation of these constraints into the portfolio optimization model and the corresponding constraint handling techniques was provided. Clearly, EF_{known} attained were affected by the inclusion of these constraints, depending on its stringency. Nevertheless, the empirical studies illustrated the viability of the proposed evolutionary approach in attaining feasible EF_{known} under the various constraints. Further studies based on proper statistical tests are necessary for the proper evaluation of the constraint handling capability of the algorithm in term of its effectiveness and efficiency. This will be reserved for future work, as currently, there are no other proposed MOEA that can operate under these constraints.

Chapter 6

Investigating Technical Trading Strategies via EMOO

6.1 Introduction

The development of technical trading strategies (TTS) has always been an important financial subject, garnering vast amount of interest from researchers for many years. Essentially, TTS are trading rules based on technical analysis - forecasting of future market movements based on the past history of market actions. However, this methodology directly contradicts the Efficient Market Hypothesis [78], which states that future market prices are completely random and thus unpredictable since all the information available is already being reflected in the current market prices. In more technical terms, the stochastic movement of markets prices is purely Markovian [136]. As such, economists have always been highly skeptical over TTS in general [179].

Nevertheless, TTS are still widely used by professional traders [88]. For example, more then 90% of the traders in London adopt technical analysis for financial forecasting [236] and a survey based in Hong Kong further underlined the popularity of technical analysis, especially in short time horizons [156]. Furthermore, recent empirical studies have shown that the market is less efficient than was originally believed. Li and Tsang [148] classified these supporting evidences into two main categories, namely systematic dependencies in security returns [33, 114, 153] and excess returns earned by the technical rules [27, 115, 146, 157, 247]. These empirical studies are realizable due to the rapid development of communication and trading facilities over the past two decades, allowing financial markets to be scrutinized in greater depth than is previously impossible. The availability of high-quality and high-frequency data enabled TTS to be evaluated more accurately as compared to traditional methods of performance evaluation [150]. Overall, these various factors have further motivated academic interest in the development of TTS lately.

Early related works mainly revolved around the evaluation of ad-hoc specified TTS that are already widely used by traders [3, 27, 79, 78]. However, these empirical studies were most likely to be spurious as such practices represent a form of data snooping due to the inherent bias in ex-post evaluation [195]. The more appropriate alternative is to develop TTS on an ex-ante premise, for instance, evolutionary computation, a type of stochastic search technique that is widely adopted due to its capability in dealing with highly complicated search space.

One of the earliest works in evolutionary TTS (ETTS) was done by Allen and Karjalainen (1995, 1999), where genetic programming was applied to generate technical trading rules in the stock market. Subsequently, many approaches based on evolutionary computation have been proposed and applied successfully to various financial data. In most existing works, the returns generated are usually used as the sole fitness measure, without accounting for the associated risk involved. However, such an approach is inadequate as TTS generally spend less time in the market and its returns are less volatile as compared to the buy-and-hold strategy [5]. Furthermore, it fails to account for the different degrees of risk averseness in every individual, corresponding to different preferences between risk and returns. Neely [178] explicitly asserted the importance of risk adjustment for the evaluation of TTS and the measure of their consistency with market efficiency [195, 217, 116, 130, 28, 67].

Ideally, TTS should have high profitability with the minimal risk possible. Unfortunately, these two criteria are inherently conflicting by virtue of the risk-returns tradeoff, where higher returns can be rendered only when subjected to a higher possibility of being lost. Hence, given the underlying nature of this optimization problem, it will certainly be instructive to cast it directly into the multi-objective domain, where the risk and returns of TTS are optimized concurrently. As such, this outlines the primary motivation for this chapter, where a multi-objective evolutionary platform will be constructed to investigate TTS from such a context. The evolutionary platform will maximize the total returns as per existing single-objective-based approaches, and concurrently minimizes risk, which is measured here by the proportion of trading period in the open position. Furthermore, the building blocks of the TTS will primarily comprise of popular technical indicators used commonly in real-world practices, which allows the examination of their trading characteristics and behaviors on the multi-objective evolutionary platform.

The remainder of this chapter is organized as such. A brief discussion on technical trading strategy will be provided, followed by a formal introduction to the multi-objective evolutionary platform. Section 6.4 presents the extensive simulation result and analysis, focusing on the insights achievable under a multi-objective formulation of this problem and the investigation on the trading characteristics of the popular technical indicators used in real-world practices.

6.2 Technical Trading Strategies

Traders can be broadly classified into two main categories, namely fundamentalists and technicians. Although both ultimately aim to forecast market price movements, their approaches and underlying principles differ greatly. Fundamentalists will analyze the market forces of demand and supply to determine its intrinsic value and enter (exit) the market if it is below (above) its intrinsic value, which is a sign of undervaluation (overvaluation). In stark contrast, technicians completely ignore the market fundamentals and decide solely based on market action i.e. the past history of market prices and trading action. Nevertheless, Murphy [177] highlighted that market prices tend to lead the known fundamentals, as price movements are usually triggered by unforeseen events. As such, usually by the time the price movements is explicated, the technician are already reaping the rewards for their early entry and analyzing for exit signal. Thus, technical analysis has been the more popular choice among traders, especially in short term trades [88, 156, 236].

Technical analysis itself can be broadly classified into two main categories: subjective and objective [246]. Subjective technical analysis or chart analysis focuses on the identification of specific visual patterns in the price history that corresponds to favorable market movements. However, as the visual interpretation of patterns varies between individuals, it is difficult to formalize its associated set of trading rules. On the other hand, objective technical analysis comprises of well defined rules or indicators that generate trading signal on whether to buy, sell, hold or wait. As such, their accuracy can be indisputably quantified. The ETTS considered in this chapter will be solely comprised of indicators based on objective technical analysis.

Broadly, these technical indicators (TI) can be defined as such,

$$TI : \{P_t, ...P_{t-n+1}\} \rightarrow [-1, 1] \tag{6.1}$$

where $TI : \{P_t, ...P_{t-n+1}\} = -1$ and $TI : \{P_t, ...P_{t-n+1}\} = 1$ corresponds to a sell and buy trading signal respectively based on the market action, P from the current time, t to a backward period of length, n. In general, P can refer to the closing prices, highest or lowest daily prices, volume traded or etc. depending on the type of TI used. While the intervals between consecutive P, could range from daily to longer durations like weekly, monthly or even yearly [13, 111] daily trading period will be considered in this chapter.

In general, a trading agent can enter the market in a long position, where the agent purchases and owns the financial asset and will profit if the price of the asset goes up, or in a short position, which refers to the practice of selling asset not owned by the seller, in the hope of repurchasing them later at a lower price. This is done so as to profit from an expected decline in price of the financial asset. The timing and type of market entry will ultimately depends on the overall trading decision of a TTS, which is in turn based on the individual signals generated from its constituent TI [54, 177]. In this chapter, for a TTS compromising of a set of m TI, the various trading signals from the TI will be combined via a weighted sum and the resultant trading decisions, D, are defined as follows,

$$D : \{TI_1, ..., TI_m\} \rightarrow \begin{cases} \text{Strong Buy if} & TI_{Buy\_high} < D \leq 1 \\ \text{Weak Buy if} & TI_{Buy\_low} < D \leq TI_{Buy\_high} \\ \text{Hold if} & TI_{Sell\_low} < D \leq TI_{Buy\_low} \\ \text{Weak Sell if} & TI_{Sell\_high} < D \leq TI_{Sell\_low} \\ \text{Strong Sell if} & -1 \leq D \leq TI_{Sell\_high} \end{cases} \tag{6.2}$$

where $-1 < TI_{Sell\_high} < TI_{Sell\_low} < 0 < TI_{Buy\_low} < TI_{Buy\_high} < 1$ represents the four thresholds that governs the traders decision with respect to the current weighted trading signal from the various TI. Specifically, the agent will enter the market in a long (short) position when the decision is strong buy (strong sell) and exit when the signal weakened to weak buy (weak sell) or worse. Also, under this definition, it is possible for an agent to switch position instantaneously, for example, an agent in a long position, originally due to a strong buy, will switch directly to a short position if D suddenly drops below lower than TI_{Sell_high}.

The following is a brief description of some of the TI widely used by real-world traders that will be used subsequently as the building blocks for the TTS. Moving average (MA) is simply the weighted average of a certain period of data and for example a simple 10-day MA of the closing prices is calculated by adding up the closing prices for past 10 days and dividing the total by 10. Other types of MA include weighted MA and exponential MA, which adopts different weight coefficients. This relationship can be generalized as follow:

$$MA(t,n) = \frac{\sum_{i=t-n+1}^{t} w_i p_i}{n} \tag{6.3}$$

where p_i and w_i refer to the closing price and its corresponding weight at time i, while t and n denotes the current time and the length of the period considered respectively

MA is typically used to detect the underlying trend direction and provide the relevant trading signal. The most common approach is the double-crossover method as formulated in (6.4), where two MA of different periods, n_1 and n_2, are considered, where $n_1 < n_2$. Basically, if the short-term (n_1) MA is larger then the long-term (n_2) MA, it corresponds to an upward trend in the price, hence generating a buy signal and vice versa. Multiple MA crossover signals are sometimes considered also to act as alert and conformation signal (Murphy, 1999).

$$TI_{MA}(n_1, n_2, t) = \begin{cases} 1 & MA(t,n_1) - MA(t,n_2) > 0 \\ -1 & MA(t,n_1) - MA(t,n_2) \leq 0 \end{cases} \tag{6.4}$$

Popular period combinations for the double crossover method are 5-20 and 10-50, though, other period has been considered also. A shorter MA, which is generally more sensitive, will trade more actively. Although earlier trading signals will thus be generated, resulting in more profitable trades, this will come at the expense of higher transaction cost and increase the likelihood of false signal. As such, the optimal parameter setting should allow the TI to possess a certain degree of sensitiveness so as to react immediately to market movements and is yet insusceptible to false signals.

Relative Strength Index (RSI), a popular price momentum oscillator, measures the strength of market movement by comparing the magnitude of its recent gains to the magnitude of its recent losses and it can be mathematically expressed as such,

$$
\begin{aligned}
RSI(t,n) &= 100 - \frac{100}{1+RS}, RS = \frac{AG}{AL} \\
&= \frac{100 \times AG}{AG + AL}
\end{aligned}
\tag{6.5}
$$

where RS is calculated as the ratio of two exponentially smoothed moving averages, $\frac{AG}{AL}$. AG and AL are respectively the average price gain and price drop from the current time, t to a backward period of length, n. Similarly to MA, a shorter time period will result in oscillations of higher frequency and amplitude, increasing its sensitivity to market movements.

RSI will typically oscillate within the range between 0 and 100, reflecting the market condition and the popular interpretation of RSI is to look for oversold levels below a specified low threshold (Low) and overbought levels above a specified high threshold ($High$), which can be formalized into the technical rule in (6.6). Trading signal will be generated when the RSI exceeds either threshold, otherwise it will correspond to a holding signal of 0.

$$
TI_{RSI}(n,t,High,Low) = \begin{cases} 1 & RSI(t,n) < Low \\ 0 & Low < RSI(t,n) < High \\ -1 & RSI(t,n) > High \end{cases}
\tag{6.6}
$$

Stochastic oscillator (SO) is a momentum or price velocity indicator that measures the position of a stock compared with its most recent trading range. Specifically, it measures the relationship between the closing price, CL of a stock and its highest high, HH and lowest low, LL from the current time, t to a backward period of length, n. The underlying rationale is that closing prices near the top of the range implies accumulation (buying pressure) and those near the bottom of the range indicate distribution (selling pressure). As such, reading below the specified low threshold or above the specified high threshold corresponds to an oversold and overbought market; hence the appropriate decision signal will be generated. This raw stochastic value is denoted as $\%K$, which is then smoothed with a simple moving average to produce $\%D$. There are several ways to interpret SO and one of the popular methods is described as follows. Basically, $\%D$ will be the 3 day moving average of $\%K$ and ,similar to RSI, trading signal will be generated when both $\%D$ and $\%K$ exceeds either threshold, otherwise it will correspond to a holding signal of 0.

$$\begin{aligned} \%K(t,n) &= 100 \times \frac{CL(t) - LL(t,n)}{HH(t,n) - LL(t,n)} \\ \%D &= \text{3 Period Moving Average of } \%K \end{aligned} \tag{6.7}$$

$$TI_{SO}(n,t,High,Low) = \begin{cases} 1 & \%K, \%D < Low \\ -1 & \%K, \%D > High \\ 0, & otherwise \end{cases} \tag{6.8}$$

While the various TI, discussed so far, have been used extensively in the financial market as a decision tool for investors or by economists to explain market phenomena, their underlying characteristics have not been fully explored before in the context of evolutionary platform. As such, the multi-objective evolutionary platform that will be introduced in the next section will evolve TTS based on these TI as the building blocks and investigate their trading characteristics, particularly their frequency in generating trading signals and their level of participation in the market.

6.3 Multi-Objective Evolutionary Platform for ETTS

Evolutionary computation is a class of stochastic search technique that has been gaining significant attention from the research community in the recent years for its success in solving real-world problems that are inherently complex with various competing specifications. The EA paradigm is largely inspired by the biological process of evolution, where potential solutions are encoded as chromosomes to epitomize the mechanics of DNA blueprint of living organisms, so as to allow the inheritance of desirable properties to offspring solutions and the propagation of information through genetic variation. The primary advantage of ETTS is that a resolute definition of the general form for the trading rules is not required and the search can be conducted efficiently in a nondifferentiable space of rules [179] on an ex-ante approach. This section presents the multi-objective evolutionary platform that will be used for the optimization of ETTS. The various features of the evolutionary platform will be introduced in turn before describing its overall algorithmic flow.

6.3.1 Variable-length Representation for Trading Agents

Depending on the representation and the evolutionary operators, evolutionary computation can be further classified into genetic algorithm, genetic programming, evolutionary strategies and evolutionary programming, with the former two being the more popular approach for the optimization of ETTS. The main difference between genetic algorithm and genetic programming lies in their representation. The former adopts pseudo-chromosomal (binary) strings to encode the information describing the underlying ETTS. The encoded information can be a masking string to include/exclude the use of certain TI [136, 245] or a direct parameter encoding of its constituent TI [119, 150, 151, 180, 210, 211]. On the other hand, genetic programming uses hierarchical variable-length strings symbolizing decision trees. The non-terminal nodes could be arithmetic, Boolean or conditional function and the terminal nodes could be variables or constant that serves as arguments of the functions [5, 39, 88, 148, 179].

The former representation is simple and straightforward and because of the fixed structure, the underlying trading rules are easily interpretable. However, since the chromosomes are constrained to certain pre-defined structures, the novelty of the TTS evolvable will be limited. While this is not an issue in the tree-based representation in genetic programming, the complexity of its search space might be too high for efficient optimization and the evolved solutions are often plagued with redundancy [5, 91]. Considering their fair share of advantages and limitations, the chromosomal representation adopted actually represents a hybrid between these two representations.

In real-world practices, technical investors usually based their trading decision on a set of TI with varying degree of importance. Their parameters will consistently be tweaked and tuned based on the traders experience and their past performance of the corresponding TI. To emulate such characteristics in the evolutionary platform postulated, trading agent are modeled as a set of decision thresholds and TI of different weights and parameters (defined in (6.2)) that will govern its trading activity, as illustrated in Figure 6.1. Adopting such a variable-length chromosomal representation [38, 234, 235], TI could be added and removed from the trading agent during the evolutionary process to adapt to the market conditions. Apart from possessing such flexibility, the underlying trading rules associated with each trading agent are comprehensible.

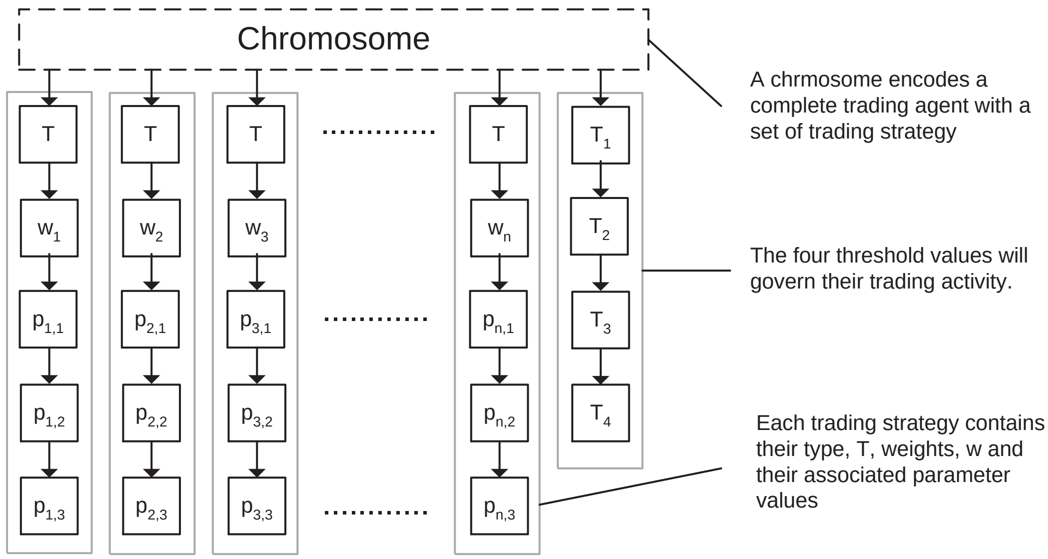


Figure 6.1: Variable-length chromosomal representation for the trading agents, which essentially comprised of a weighted combination of a set of commonly-used TI in real practices.

The description for the various genes in the variable-length chromosome is summarized in Table 6.1. The trading agents will build their strategies based on these three different TI and due to the variable-length chromosomal structure, TI of the same type can exist together in a single chromosome. Essentially, the optimization process involved finding an optimal combination of TI with appropriate parameters and weights.

6.3.2 Objective Functions

In related literature, the fitness/optimality of ETTS is either measured by the accuracy of the predictions made [148, 158, 233] or solely based on its profitability. As the former is more applicable for classification problem, the latter represents the more intuitive choice for performance evaluation. Even so, the latter could be measured in different ways, for example, the total asset, namely the

Table 6.1: Gene Description of the various parameters (general and TI-specific) being optimized.

Technical Indicator	Parameters	Range
MA	Short term period	[1,50]
	Long term period	[1,50]
RSI	Period	[1,40]
	RSI selling threshold	[1,50]
	RSI buying threshold	[50,100]
SO	Period	[1,40]
	SO selling threshold	[1,50]
	SO buying threshold	[50,100]
-	Weights	[0-1]
-	Decision sell threshold	[-1,0]
-	Decision buy threshold	[0,1]

available capital and the value of all holdings, at the end of the trading period [245, 127] or the area under the total asset graph during the trading period [211]. In other cases, the generated profits are directly pegged to those generated by the buy-and-hold strategy [5, 88]. However, all these measures failed to acknowledge the risk involved with the trading activity [178]. As such, performance measures like Sharpe ratio or Sterling ratio was proposed instead, which can measure the net profitability after discounting the associated risk [137, 178].

Clearly, the optimization of ETTS involves a delicate balance between its expected returns and associated risk. As such, contrary to conventional single-objective approaches, where the risk is completely ignored or the two conflicting objectives of risk and returns are combined into one single measure known as the risk-adjusted profit, this chapter will model the problem directly as a multi-objective optimization problem by simultaneously optimizing the returns and risk of the ETTS.

Considering a period of length T, corresponding to a total number of n trades, the total returns for the period is defined as such,

$$\text{Maximize } F_1 : Total\ Returns = \sum_{i=1}^{n} k \times \frac{p_{i,exit}}{p_{i,entry}} \tag{6.9}$$

$$k = \begin{cases} 1, & \text{short entry} \\ -1, & \text{long entry} \end{cases}$$

where $p_{i,entry}$ and $p_{i,exit}$ denotes the price at which the trading agent enter and exits the market respectively for the ith trade and the multiplier, k is to adjust the returns to compensate for the different type of entries i.e. a depreciation in asset prices actually corresponds to a profit for a short entry. In essence, this objective function simply measures the arithmetic total of the percentage price changes for all the trades made in the trading period. The arithmetic total is considered here instead of the mean, as a profitable trading rule may forecast rather poorly much of the time, but perform well overall because it is able to position the trader on the right side of the market during large moves [179]. Also, the use of percentage changes removes the dependence on entry prices as compared to absolute value difference. Lastly, this measure accounts entirely for the profitability and avoids the need to define the order size for each trade.

Risk is defined as the volatility or the uncertainty of the expected returns over the trading period. For instance, a TTS that yield returns ranging between 4% to 6% is less volatile then one, whose returns ranges between -40% and 50%, even though their average returns is the same. Standard risk measures like variance or semi-variance are not suitable here. The former fails to consider that investors are more averse to negative deviations about the mean returns as compared to positive returns [15]; while the latter is somehow correlated to the returns i.e. minimizing the semi-variance will indirectly maximize the total returns. In fact, preliminary investigation that considers them as the risk measure failed to obtain a Pareto front that could accurately depicts the risk-returns tradeoff.

Instead, risk will be defined here by the traders exposure to it. Specifically, it will be measured by the proportion of trading days when an open position is maintained in the market [246] and is mathematically formulated as such,

$$\text{Minimize } F_2 : Risk\ Exposure = \frac{1}{T}\sum_{i=1}^{n} t_{i,exit} - t_{i,entry} \tag{6.10}$$

where $t_{i,entry}$ and $t_{i,exit}$ denotes the time at which the trading agent enter and exits the market respectively for the ith trade and T refers to the total length of the trading period. Essentially, staying longer in the market corresponds to a higher exposure to risk like catastrophic events and market crashes while a shorter period, which is associated with lower risk exposure, will corresponds to higher liquidity as the available capital is tied up for a lesser time. Such an optimization function will be in conflicting nature with returns, as higher total returns are usually associated with higher degree of trading activity, which naturally leads to a longer periods of open position. Even though risk exposure is being considered here, this measure will, at times, be conveniently referred to as risk here.

6.3.3 Fitness Evaluation

The fitness evaluation process is concerned with calculating the total returns and risk exposure associated with each trading agents in the evolving population. During the stipulated trading period, each TI of the trading agent will generate trading signals on a daily basis based on the current and historical market actions. The overall trading decision to buy, sell or hold is obtained by considering the weighted sum of the various individual signals and the decision threshold of the trading agents. The corresponding fitness values of the trading agent can be subsequently calculated once the trading schedule is determined.

For a clearer illustration, lets consider an instance of the variable-length chromosome (Figure 6.2) being applied to a hypothetical price series (Figure 6.3) comprising of 250 trading days. The TTS involved comprises of the three different TI and their respective trading signal within the entire trading period is illustrated in Figure 6.4. For MA in Figure 6.4(a), a buy signal of 1 will be generated if the difference is positive and conversely, a sell signal of -1 will be generated, when the MA difference i.e. $MA(t, n_1) - MA(t, n_2)$ falls below zero. For RSI and SO, buy signal and sell signal will be generated when they exceed the buying threshold or fall below the selling threshold respectively. Otherwise, it will correspond to a hold signal of 0.

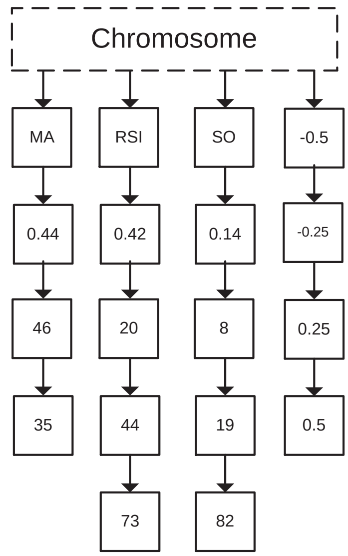


Figure 6.2: An instance of the variable-length chromosome comprising of the three different TI i.e. MA, RSI and SO.

Clearly from the plots in Figure 6.4, trading signals from the various TI are different in terms of the trade frequency and duration. MA tends to generate longer buy and sell signal due to the nature of its definition, while SO generated shorter buy and sell signal but at a higher frequency. Also, due to the high RSI buying threshold, no buy signals were generated from RSI. A more in-depth analysis on the characteristic of the various TI will be conducted in section IV.

The overall decision signal, which is the weighted sum of its constituent trading signals, is illustrated in Figure 6.5. The trading agent will enter the market in a long position, whenever the decision signal goes above T_{Buy_high} of 0.5, denoting a strong buying signal. Correspondingly, it will exit anytime the decision signal falls below T_{Buy_low} of 0.25, as the buying signal has weakened. Short sell, which is considered also in this model, will be executed vice versa, based on T_{Sell_high} and T_{Sell_low}.

It should be highlighted that, for simplicity in the trading model, it is assumed that the trading environment is a discrete and deterministic liquid market, where the price is unaffected by the agents actions. The trading schedule of the agent is tabulated as follow. Altogether, the agent performed seven trades within this period. While there are no limits on the number of trades conducted, each complete trade will be subjected to a fixed transaction cost of 0.5% to simulate brokerage charge,

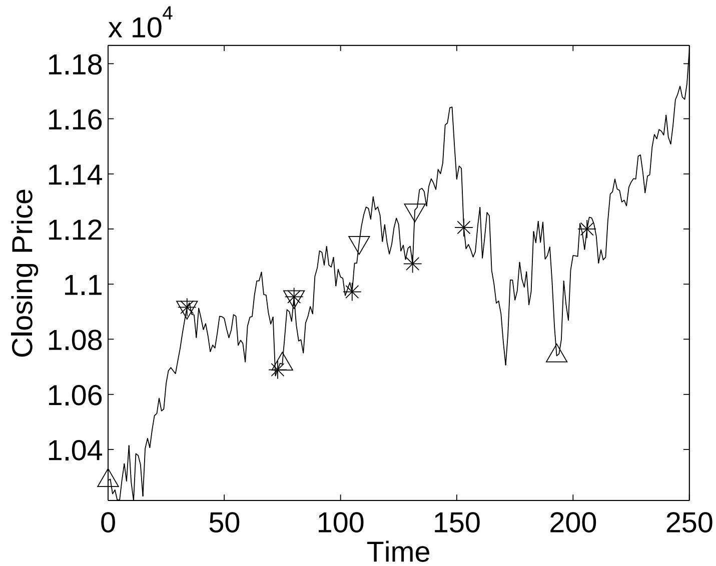
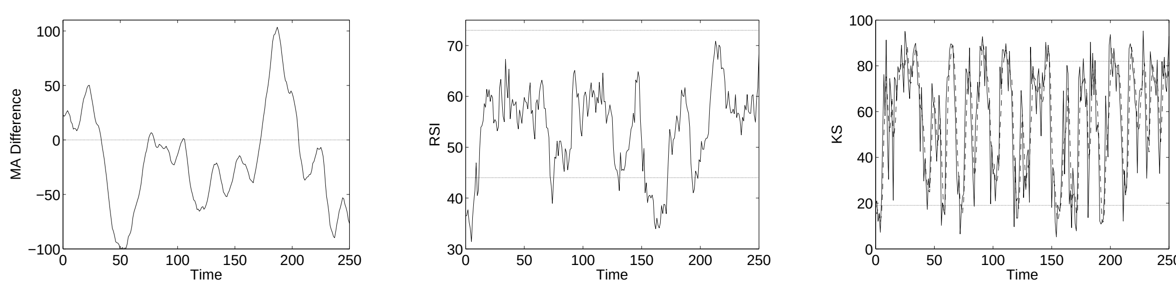


Figure 6.3: Hypothetical price series comprising of 250 trading days. Trading activity determined in Figure 6.5 is included where upward triangle, downward triangle and asterisk denote long entry, short entry and exit respectively.



(a) (b) (c)

Figure 6.4: Traces of the trading signals generated by the various TI over the trading period. The respective thresholds of RSI and SO are denoted by the horizontal dotted lines.

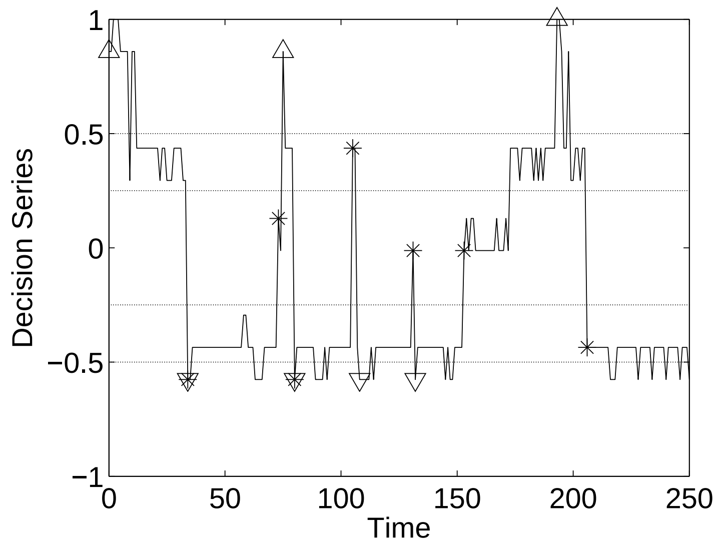


Figure 6.5: Trace of the overall trading signal (Upward triangle, downward triangle and asterisk denote long entry, short entry and exit respectively). The respective thresholds are denoted by the horizontal dotted lines.

interest rate, liquidation costs or other form of costs.

Table 6.2: Trading Schedule of the agent in Figure 6.2 and the calculation of its total returns and risk exposure with the trading period.

Trade No.	Trade type	Entry Time	Exit Time	Trading Period	Entry Index	Exit Index	Returns (%)	Net Returns (%)
1	long	0	34	34	10287	10916	6.11	5.61
2	short	34	73	39	10916	10689	2.08	1.58
3	long	75	80	5	10710	10954	2.28	1.78
4	short	80	105	25	10954	10972	-0.16	-0.66
5	short	108	131	23	11151	11074	0.69	0.19
6	short	132	153	21	11269	11206	0.56	0.06
7	long	193	206	13	10739	11200	4.29	3.79
		Total	Proportion	$\frac{160}{250}$	Total	Returns	15.85	12.35

The calculation of returns and risk associated with the trading agent shown in Figure 6.2 is tabulated in Table 6.2. As mentioned earlier, the arithmetic total of each percentage gains net of transaction cost and the proportion of days in open position are used to quantify the total returns and risk exposure respectively. In this particular example, the corresponding fitness values are calculated to be 12.35% and 0.64 respectively.

6.3.4 Pareto Fitness Ranking

Evolutionary optimization of TTS is cast as a multi-objective problem in this chapter, which involves the maximization of the total returns and minimization of risk exposure. In contrast to single-objective optimization, the optimal solutions to a multi-objective optimization problem exist in the form of alternate tradeoffs known as the Pareto-optimal set. Each objective component of any non-dominated solution in the Pareto-optimal set can only be improved by degrading at least one of its other objective components. The Pareto-optimal set when plotted will constitute the risk-return tradeoff or Efficient Frontier as illustrated in Figure 6.6.

Each point denotes a TTS evolved by the MOEA and the black and gray circles represents non-dominated and dominated solutions respectively. The former set is the Pareto optimal solution as their returns cannot be improved further without compromising risk. In the context of single optimization of ETTS where returns is the sole priority, the evolutionary process will ultimately drive the solutions towards the extreme point B. This is not applicable for conservative investors, who prefer lower risk as compared to higher returns. Point A represents the extreme case of a conservative investor with zero returns due to an empty trading schedule.

In the total absence of information regarding the preference of objectives, the Pareto ranking scheme is considered to represent the fitness of each trading agent in such a context. Specifically, this scheme assigns a default minimal ranking for all the non-dominated solutions, while the dominated solutions will be ranked according to how many other solutions in the population dominate them. The rank of a solution, i, in the population is

$$Rank(i) = 1 + n_i \tag{6.11}$$

where n_i is the number of ETTS dominating the ith ETTS in the population pool.

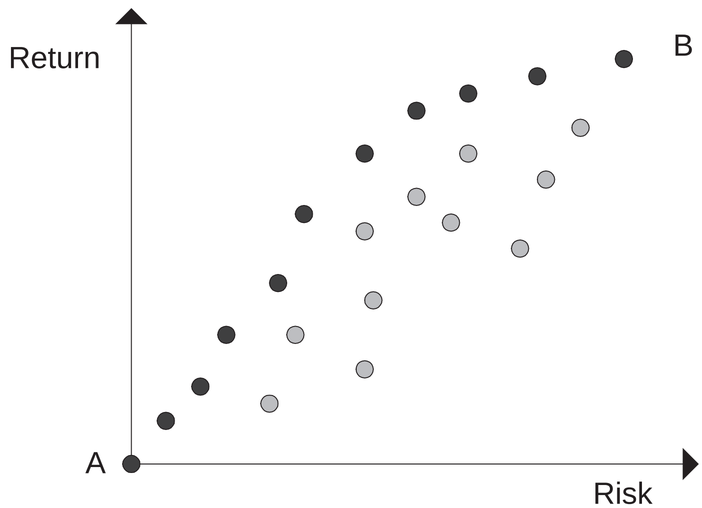


Figure 6.6: Illustration of the risk-returns tradeoff.

6.3.5 Variation Operation

As standard chromosomal representation with well-established variation operators were not considered here, specific operators need to be designed for this purpose. In evolutionary algorithm, good combinations of genes are exchanged between different chromosomes of the population via crossover operators. The crossover operator adopted for the variable-length chromosome is illustrated in Figure 6.7. Essentially, the crossover operation involves combining the TI for the two parent chromosomes and randomly distributes them amongst the two child chromosomes. The threshold values will be inherited directly during the process.

The crossover operation will be complemented by a multi-mode mutation operator [234, 235] in allowing a larger search space to be explored. The primary motivation for such a operator is that the variable-length chromosome adopted has varying hierarchy, in terms of the different type of indicators and their corresponding weights and parameters. As such, the multi-mode mutation operator is to cater for such data structure and allow the variable-length chromosome to be altered at varying levels. Each mode represents varying degree of perpetuation in the search space and thus signifies different exploration and exploitation efforts. The various modes are as such:

1. *Indictor level*: At random, an existing TI is being removed from the trading agent or a random

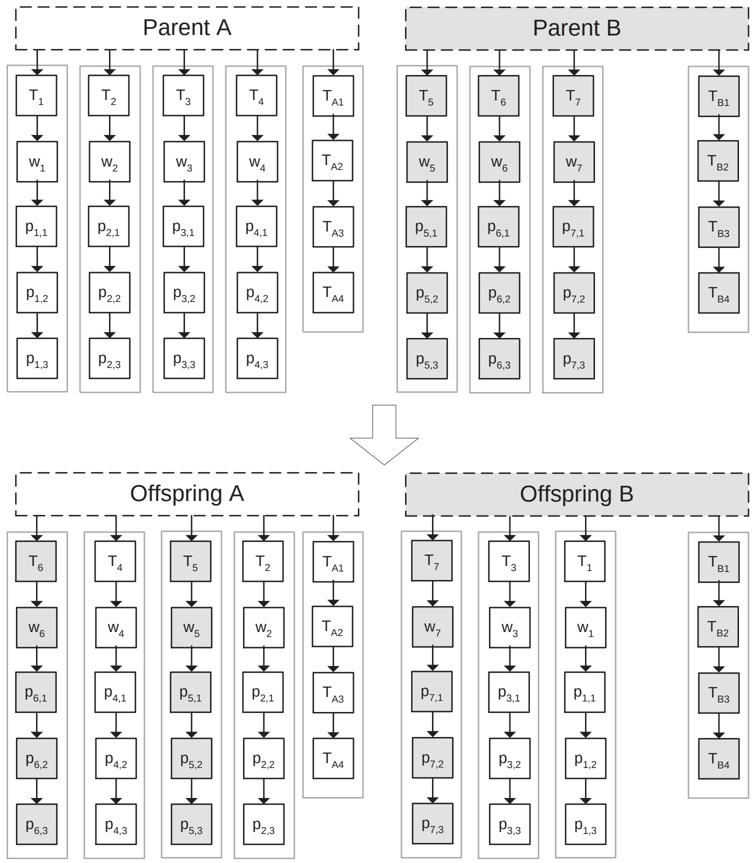


Figure 6.7: Illustration of the trade-exchange crossover.

indicator is being initialized and added to it.

2. *Parameter level*: A TI is being chosen at random and its parameters are subjected to Gaussian mutation

3. *Threshold level*: The various TI remain unchanged, while the four threshold values are subjected to Gaussian mutation

These three different modes will be invoked randomly during the mutation operation.

6.3.6 Algorithmic Flow

The algorithmic flow of the multi-objective evolutionary platform is shown in Figure 6.8. At the start of the algorithm, the decision signals of the TI under all possible parameter configurations are pre-calculated. This allows direct access of this information during algorithmic runtime, which will speed up the computation time required.

- *Initialization*: The algorithm maintains a fixed size population throughout the evolutionary process. During the initialization process, trading agents will be randomized until the population is filled. Specifically, it will involve randomly generating a number of TI within a predefined range of random weights and parameters

- *Elitism*: Although multi-objective evolutionary algorithms have been implemented in many different ways, most current state-of-the-art works in general encompasses some form of elitism in both the archiving and selection process. A fixed size archive is used to store the non-dominated solutions discovered during the evolutionary process. The archive is updated every generation, where agents that are not dominated by any members in the archive will be added into the archive and any members in the archive that are dominated by this new agent will be removed. The archive helps to ensure convergence by preventing the loss of good solutions due to the stochastic nature of the evolutionary process. In the selection process, elitism is implemented by selecting individuals to a mating pool through a binary tournament selection of the combined archive and evolving population. The selection criterion is based on Pareto ranking and in the event of a tie, the niche count will be employed. The mechanism of niche sharing is used in the tournament selection as well as diversity maintenance in the archive.

- *Reinsertion*: Randomly generated trading agents are added to the evolving population every generation to complement the variation operation. This allows greater exploration of the search space and prevents premature convergence by introducing genetic materials that are not formally present in the initial gene pool. This approach is similar to immigration [24, 211], but the latter directly replaced the mediocre proportion of the populations by these new solutions. Such an approach is avoided as the ordering of solutions in multi-objective optimization is not so straightforward.

After one complete generation, the evolutionary process will repeat until a predefined number of generations are reached.

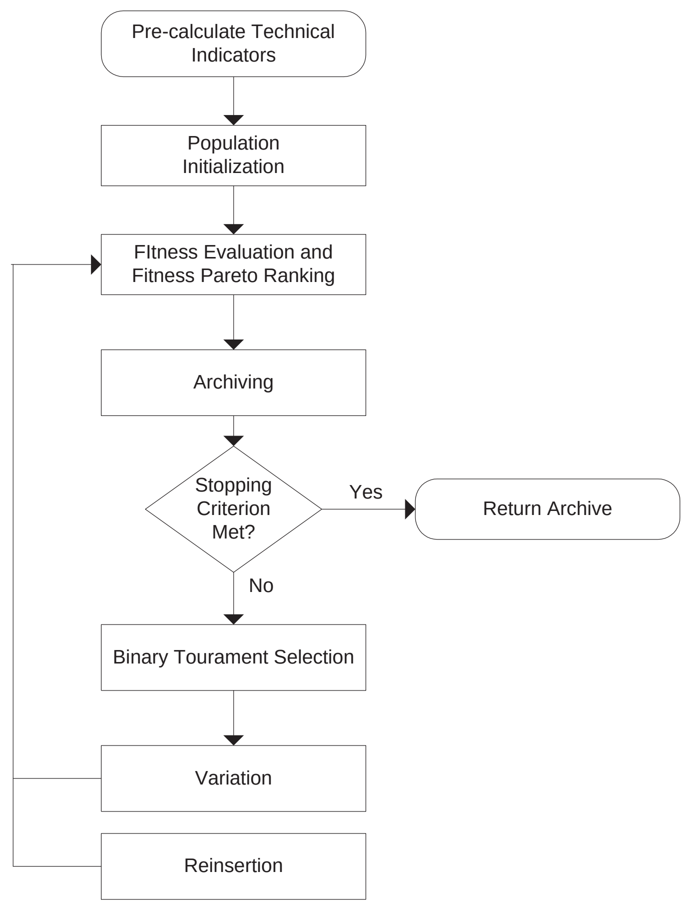


Figure 6.8: Algorithmic flow of the multi-objective evolutionary platform.

6.4 Simulation Result & Discussion

The viability of the multi-objective evolutionary platform and the TTS evolved will be studied in this section. The effectiveness of hybridizing multiple TI to form TTS will be investigated first, followed by an assessment of the generalization performance of the ETTS evolved. The performance of the ETTS will be compared against the buy-and-hold strategy, which is essentially a long term investment strategy in which stocks are bought and then held for a long period regardless of the market's fluctuations. The argument for this strategy is actually the Efficient Market Hypothesis [78], whereby if every security is fairly valued at all times, then there is really no incentives to trade. This underlying principle behind the buy-and-hold strategy is a stark contrast as compared to that of technical analysis.

The financial data considered here are the daily trading data of the Straits Times Index (STI), a market value-weighted stock market index that is based on the stocks of 50 representative companies listed on the Singapore Exchange. A total of 3368 trading days were considered, which span from the period 11-08-1992 to 30-12-2005. However, the actual number of trading data used will differ in the various simulations.

The parametric configurations of the multi-objective evolutionary platform outlined in section III are summarized in Table 6.3. These parameters have been selected based on a series of preliminary investigation and parameter tuning. A reinsertion ratio of 0.1 denotes that 10% of the children chromosome at every generation will comprise of randomly generated trading agents, with the remaining coming from the variation operation. 20 independent runs were made for each simulation with each set of runs having the same random seed to ensure the same initial population.

Table 6.3: Parameter settings of the multi-objective evolutionary platform used in the simulations.

Parameters	Values
Population Size	100
Generation	1000
Crossover Rate	0.8
Mutation Rate	0.1
Reinsertion Ratio	0.1
Maximum Number of Technical Indicators in a Trading Agent	10
Number of Simulation Runs	20

6.4.1 Performance Comparison between Individual TI and Hybrid TI

Many different types of TI had been considered in previous related works on ETTS and the exact quantity constituting the trading agent can easily vary from one [119] to even hundreds [136]. Despite so, the effects of hybridizing several TI as opposed to applying them individually in constructing TTS have never been studied in-depth before. Thus, this section will investigate whether the hybridization of TI are synergetic and destructive in nature. For this purpose, different combinations of the three TI were considered as listed in Table 6.4 and the same evolutionary platform was adopted with the only difference being the type of TI available as the building blocks for the trading agents during the evolutionary process. For example, ETTS evolved by D1 will only comprise of MA and RSI, but not SO.

Table 6.4: Different combinations of TI used to assess the hybridization of TI in the trading agents.

Combination Description	Notation
MA only	MA
RSI only	RSI
SO only	SO
MA and RSI	D1
MA and SO	D2
RSI and SO	D3
MA, RSI and SO	ALL

The trading agents are optimized based on a 4 year financial data from the period 02-01-2002 to 30-12-2005 as plotted in Figure 6.9. 100 days of historical data prior to the first trading days are included also, as some TI need a certain amount of historical data in their calculations. Figure 6.10 plots the Pareto fronts evolved by some of the TI combinations in one of the runs. While the various solutions sets are of varying optimality in terms of Pareto dominance, they clearly illustrate the inherent tradeoff between total returns and risk exposure. Also, the trading agents evolved are able to generate high returns in open positions less than 100% of the trading period, i.e. total returns above 80% with risk exposure around 0.6 by ALL. Comparatively, the total return of the buy-and-hold strategy is only 44% during this period. Clearly in this context, the buy-and-hold strategy is suboptimal.

Considering discrete intervals of 0.1 for risk exposure, Figure 6.11 plots the distribution of the average returns and number of trading agents along the Pareto front obtained by ALL in 20 runs. The risk-returns tradeoff is again evident in Figure 6.11(a), where the average returns increases for

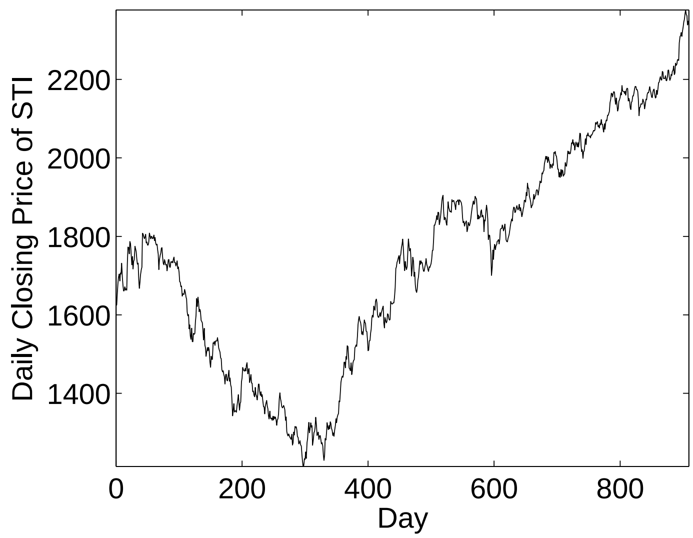


Figure 6.9: Daily closing prices of STI used for the optimization of the ETTS.

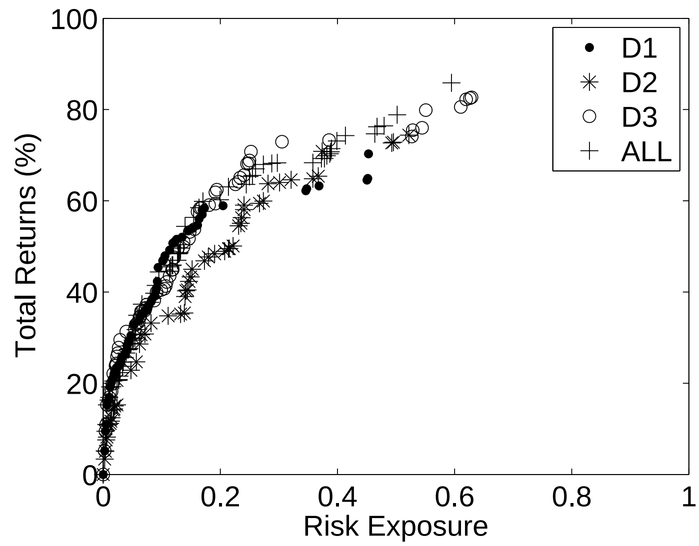
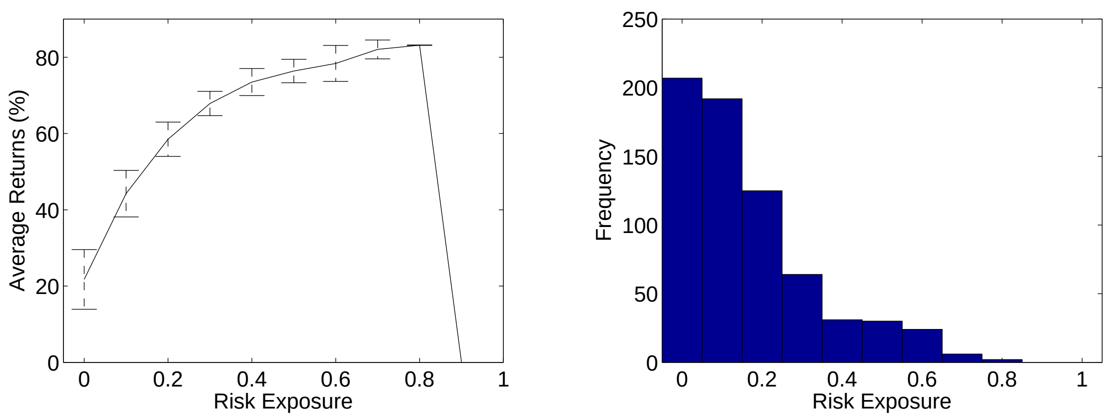


Figure 6.10: Pareto fronts obtained by the selected TI combinations in one of the runs.

higher level of risk exposure. The sudden drop in average returns after risk exposure of 0.8 can be attributed to the lack of solutions in that region, as illustrated in Figure 6.11(b). In fact, the evolved Pareto front is not uniform as there is a higher density of solutions at the lower risk-returns region. The non-uniformity could be due to the general difficulty in finding TTS that can fully exploit all the price movements and generate exceptionally high returns in the presence of transaction cost.



(a) Average Returns versus Risk Exposure

(b) Number of trading agents versus risk exposure

Figure 6.11: (a) Average returns and (b) number of trading agents in discrete intervals of Risk Exposure of 0.1 that were generated by ALL in 20 runs. The vertical line in (a) indicates the standard deviation of the returns at each discrete level of risk exposure.

Of course, a visual comparison of the Pareto front is not adequate for a complete performance assessment between the various combinations. Binary quality measures, which compare the dominance relationship between pairs of solutions sets, should be adopted [263]. For this purpose, the coverage function (C) [262] is included, which gives for a pair of solutions sets (A, B) the fraction of solutions in B that are weakly dominated by one or more solutions in A. The value C(A, B) = 1 means that all the points in B are dominated by, or equal to the points in A. The opposite, C(A, B) = 0 represents the scenario, when none of the points in B are covered by the set A. It should be highlighted that both C(A, B) and C(B, A) have to be considered for a complete performance assessment.

The coverage metrics represent quantitative measures that describe the quality of the evolved solution and they are illustrated in box plots in Figure 6.12 to provide the statistical comparison results. The thick horizontal line within the box encodes the median while the upper and lower ends denote the upper and lower percentile respectively. Dashed appendages illustrate the spread and shape of distribution and crosses represent extreme values.

From the boxplots, the hybrid combinations are clearly better than the individual combinations where the agents evolved by the former are able to Pareto-dominate a larger percentage of those generated by the latter in terms of higher returns at a lower risk exposure. Also, the performance difference between ALL and the individual TI combinations are much more significant than that of ALL and the dual TI combinations. This seems to suggest some form of diminishing marginal benefits in considering more TI in the construction of TTS. As such, in this context, excessive TI should be avoided, which could also help to maintain the complexity of the search space. Of course, further simulations involving other types of TI should be conducted to validate this hypothesis.

Amongst the individual TI, RSI has the best performance in coverage, as it is able to dominate a larger proportion of the solutions evolved by MA and SO, yet having a small proportion of its solution being dominated by them as observed from Figure 6.12(a)-(c). This relationship is similarly observed for the dual combinations, where D1 and D3 which comprised of RSI perform much better than D2. Nonetheless, statistical analysis reveals no significant performance difference between D1 and D3.

The two main goals in multiobjective optimization include proximity and diversity [62, 18], where the former describes the accuracy of the solution set and the latter measures how well the solution set is defined. While the coverage function compares the proximity relationship between the various combinations, it is also necessary to assess their diversity relationship by measuring the extent in which the optimal Pareto front is covered by the evolved solutions as in the Maximum Spread measure [261]. Of course, Maximum Spread is only applicable for benchmark optimization problems where the optimal Pareto front is known. Thus, an alternative measure is proposed here, which simply computes the area in the objective search space covered by the solution sets.

$$Spread = (return_{max} - return_{min}) \times (risk_{max} - risk_{min}) \tag{6.12}$$

The boxplots in Figure 6.13 illustrates the average spread for the various TI combinations in 20 runs. Interestingly, RSI, which attained better performance in Pareto dominance amongst the individual strategy, has the lowest spread. On the contrary, D2, which do not include RSI in its

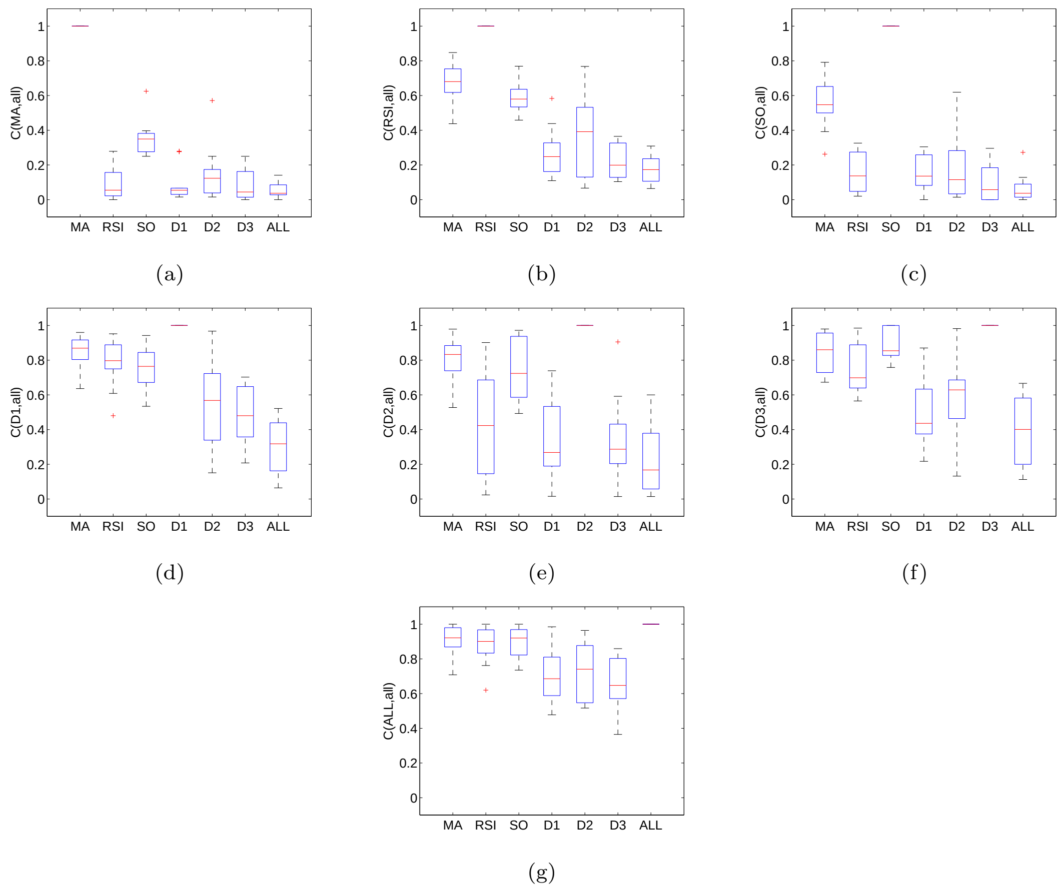


Figure 6.12: Box plots illustrating coverage relationship between the various TI combinations schemes.

composition, attained the highest spread amongst the various combinations. This seems to suggest that different TI is instrumental in attaining solutions at different regions of the risk-return tradeoff. Figure 6.14 compares the Pareto attained by RSI and D2 in one of the runs and it corresponds accurately to the results so far i.e. D2 was able to diversify over a larger area of the objective but most of its evolved solutions were dominated by RSI in the low risk-returns region.

To shed light on the underlying differences between the various TI, Figure 6.15 plots the mean number of trades versus risk exposure for the various TI combinations considered. ETTS, comprising of MA and/or SO, trade more actively as compared to those consisting of RSI. These characteristics are actually elucidated by their trading signals in Figure 6.4 also. Comparatively, SO generates buy

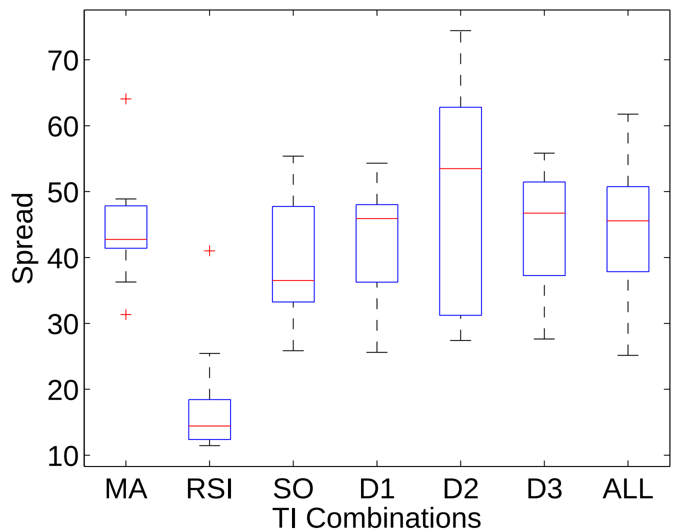


Figure 6.13: Box plots illustrating Spread obtained under the various TI combinations schemes.

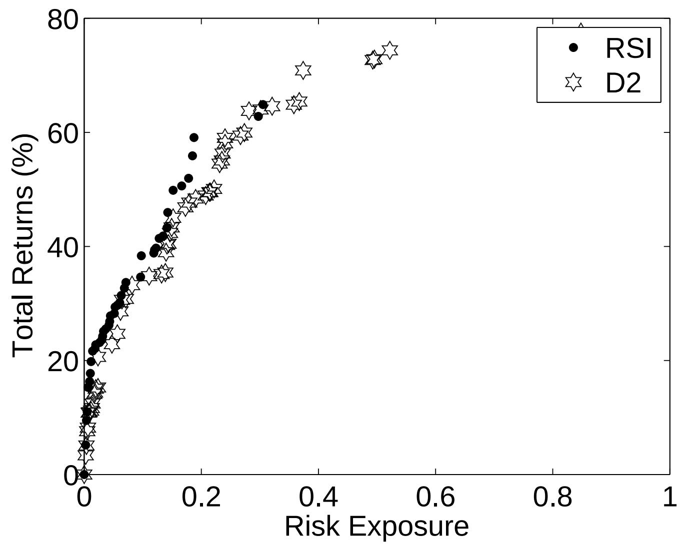


Figure 6.14: Pareto fronts obtained by RSI and D2 in one of the runs.

and sell signals at a higher frequency and MA, which, does not have any holding signal at all, tend to remain at an open position in the market. Also, it is observed that all combinations except RSI have a stable uptrend. One possible explanation is that the solutions generated by RSI are overly concentrated in the low risk region, resulting in erratic behavior in the high risk region.

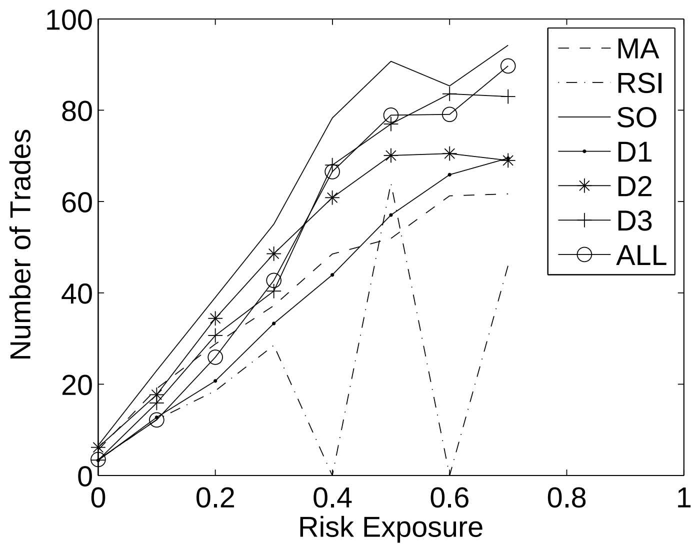


Figure 6.15: Average number of trades by the trading agents in discrete intervals of Risk Exposure of 0.1 that were generated by the various TI combinations in 20 runs.

The results so far seem to suggest that RSI aids better in low risk trading, while MA and SO are more prominent in high risk trading. This is most probably due to the latters higher tendency to generate buy/sell signal, increasing the possibilities to generate active trading schedule of greater risk exposure and returns, which will consequently results in higher spread. To further affirm the trading characteristics of the TI, it will certainly be instructive to investigate the composition of the various TI in the chromosomes evolved by ALL. Figure 6.17 plots the average weight and frequency of the various TI in each ETTS obtained by ALL. As the type of TI available is lesser than the maximum TI allowable for each agent, there could be multiple instances of the same TI in a single ETTS. Clearly, SO constitutes the largest proportion both in terms of the weights and frequency.

Lastly, to analyze how the composition of TI in the trading agents changes along the risk-returns tradeoff, Figure 6.18 plots the average weights of the various TI for each trading agent against their corresponding risk exposure. Clearly from Figure 6.18(b), there is a higher density of solutions in

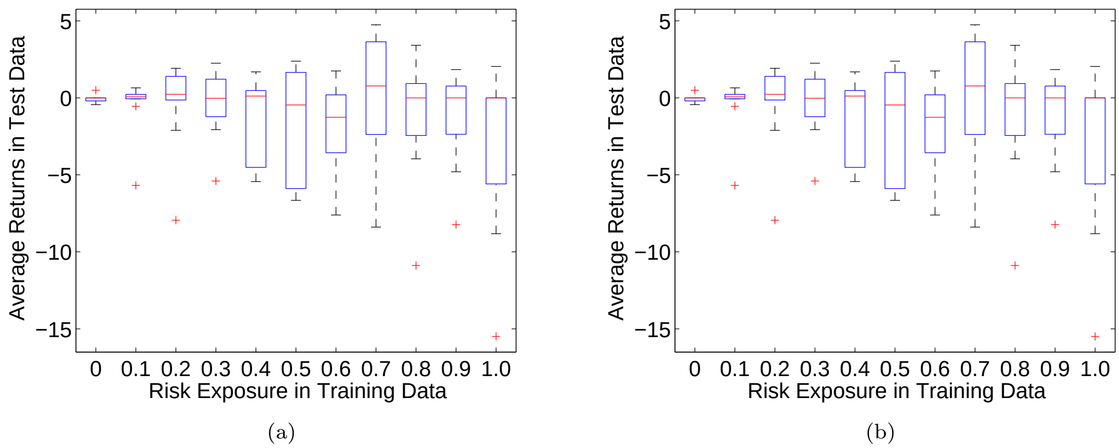


Figure 6.16: (a) Mean and (b) variance of the test returns

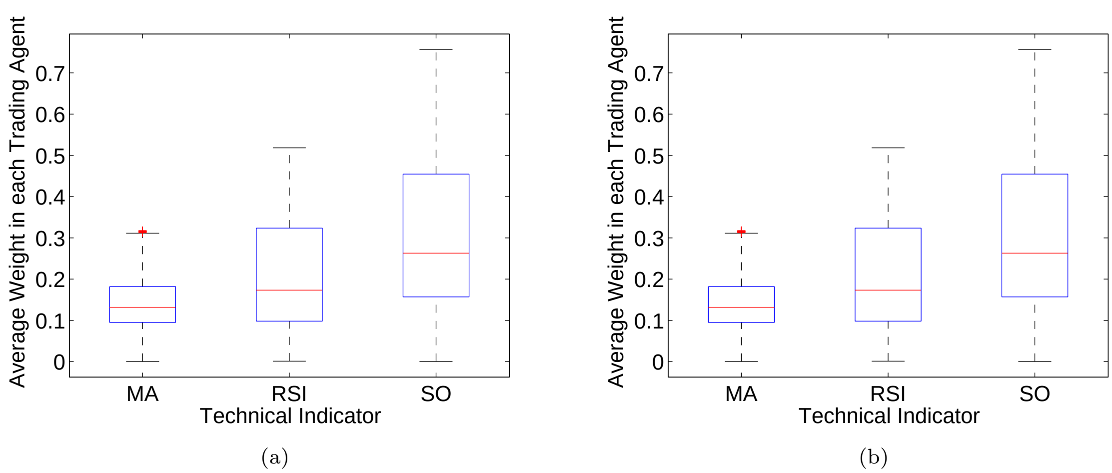
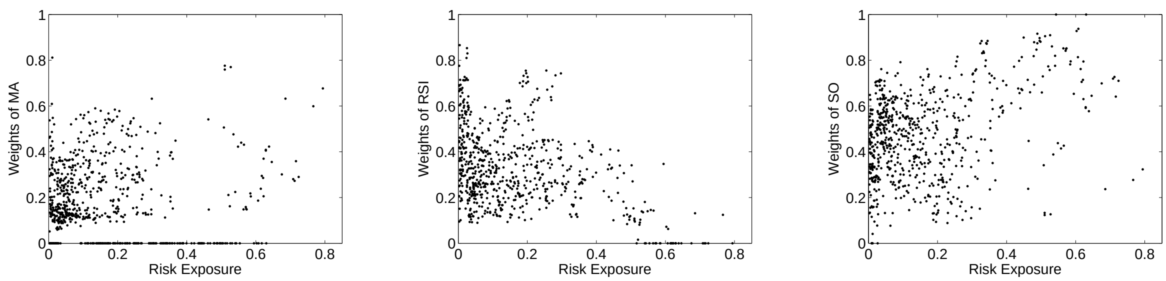


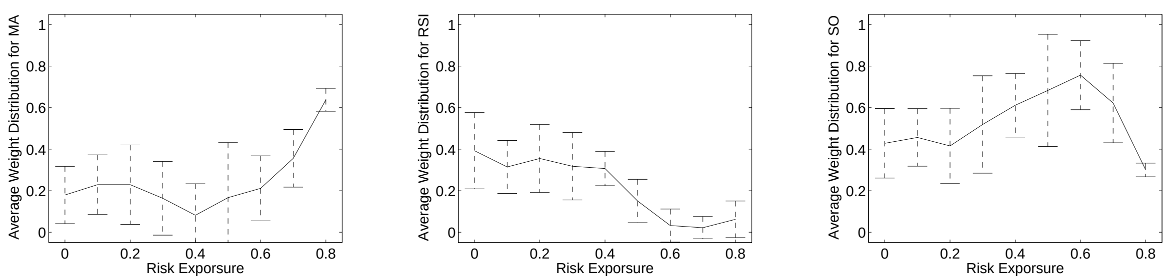
Figure 6.17: (a) Average weight and (b) frequency of the individual TI in each trading agent evolved by ALL

the low-risk region as compared to the high-risk region. This further reinforces the earlier claims that associate RSI with conservative trading schedules. To illustrate clearer the composition changes along the risk-returns tradeoff, interval of 0.1 for the training risk is considered and the mean and standard deviation of the weights at each interval is plotted in Figure 6.17. As expected, RSI, which is associated with conservative trading schedules, has higher weights at lower risk, while MA is more prominent in active trading schedules, where its average weight increases for higher risk. Lastly, Figure 6.17(c) illustrates that SO forms a significant proportion of the ETTS at the various risk level, which again could be explained by its balance between the proximity and diversity goals.



(a) (b) (c)

Figure 6.18: Average weight of (a) MA, (b) RSI and (c) SO in each trading agents versus risk exposure.



(a) (b) (c)

Figure 6.19: Statistical distribution of the average weight for (a) MA, (b) RSI and (c) SO at discrete values of risk exposure of 0.1. The vertical lines denote the standard deviation of the weight at each value of risk exposure.

The various results show that the composition of TI along the risk-returns tradeoff is related to their underlying characteristics. As such, the non-uniformity in the risk-returns tradeoff, where the diversity of solutions decreases at higher level of risk (Figure 6.10(b)), could be due also to the lack of TI that can generate highly active trading schedules. Thus, more TI should be included in future related studies. Also, it will be useful to profile these TI for a better understanding of their trading operation. This information could be useful in the development of local search operators that can exploit their underlying characteristic, so as to improve the algorithmic convergence of the evolutionary platform.

6.4.2 Correlation Analysis between Training and Test Performance

The results earlier revealed the trading characteristics of the TI constituting the ETTS. Despite the high returns generated by the ETTS at various level of risk exposure, the practicality of this approach will ultimately depend on whether these high returns can be extended to unseen trading data, which is otherwise known as its generalization performance.

To evaluate its generalization performance, the historical financial data used for the training and evaluation of ETTS earlier will be further partitioned into two independent set i.e. training and test set. During the evolutionary process, the fitness of the trading agents will be assessed based on the training set. The final solutions obtained at the terminal generation will be subsequently applied to the test set to evaluate its generalization performance, indicating its real efficiency in unseen data. The total returns and risk exposure in the training and test data will be conveniently referred as training returns, training risk, test returns and test risk respectively in the rest of the chapter.

The risk-returns tradeoff for the training and test data in one of the run is illustrated in Figure 6.20. As the TTS are evolved with respect to the training data, the risk-returns tradeoff is clearly evident in Figure 6.20(a). However, such relationship is not evident when the same set of TTS is applied to the test data. While returns of 60% are achievable at a risk level of 0.2 in the training data, losses are incurred at the same level of risk in the test data. Clearly, positive returns in the training data do not necessarily correspond to positive returns in the test data. In fact, the low correlation between training and test returns was also briefly suggested by Korczak and Lipinski [137] before, where they observed that applying a fitness measure strongly based on returns usually result in inefficiencies in test data. As such, they even suggested that profits should be restricted to post training assessment. Nevertheless, most single-objective approaches for ETTS are still based on the underlying assumption of the positive correlation between training and test returns.

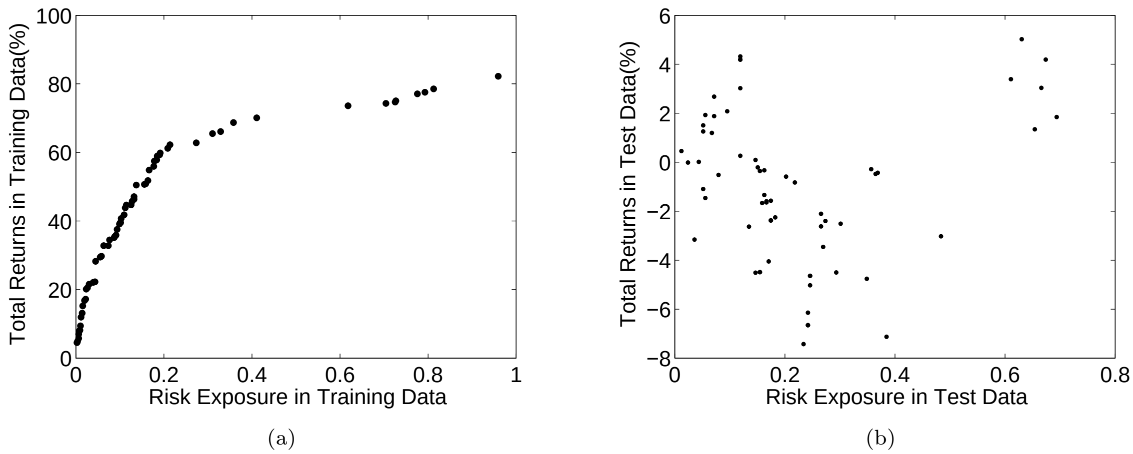


Figure 6.20: Pareto fronts obtained for (a) training data and (b) test data

Thus, before examining the generalization performance of the ETTS in proper, it is important to understand further the correlation between the performance in the training data and test data. Specifically, this refers to the correlation analysis of the following four variables: training returns, training risk, test returns and test risk.

Based on the ETTS found in 10 runs, the correlations between the various variables are plotted in Figure 6.21. Several interesting insights are revealed in these plots. The plot of training returns and training risk illustrates accurately the risk-returns tradeoff, while a noisy version can be observed in the plot of test returns and test risk, despite the low correlation between training returns and test returns.

Contrary to the assumption in single-objective optimization approach where higher training returns is associated with higher test returns, this relationship is sorely missing from these plots. Instead, higher training returns are associated with increased volatility in the expected test returns as reflected by the widening spread of the solutions. This is illustrated clearer in Figure 6.22, which plots the mean and the standard deviation (denoted by the vertical lines) of the test returns at intervals of 10 for the training returns. While the mean test returns does not increase much for larger values of training returns, there is a general increase in the standard deviation instead.

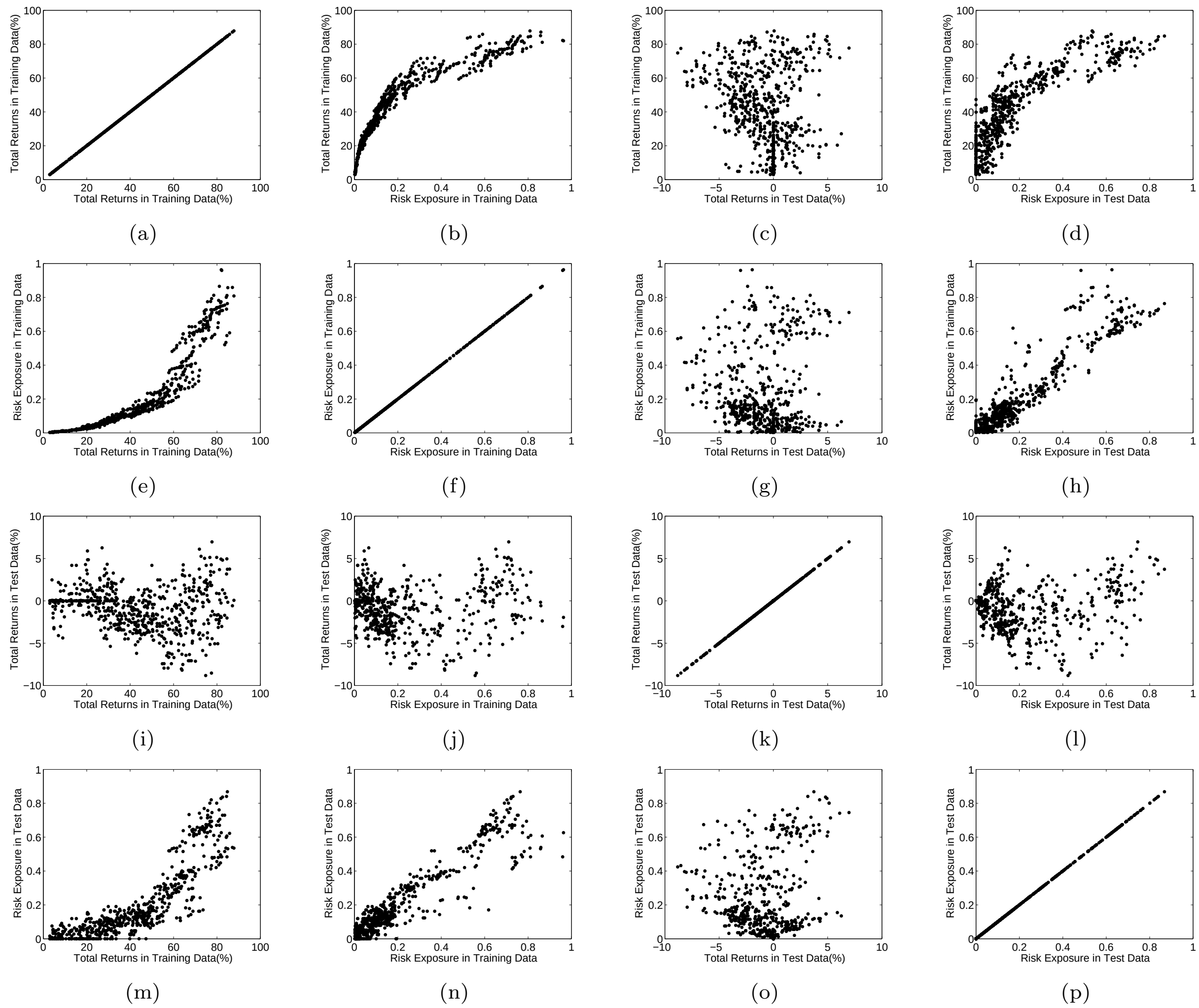


Figure 6.21: Correlation between Training Returns, Training Risk, Test Returns and Test Risk.

Noticeably, there is a positive correlation between the training risk and test risk instead, though it is slightly skewed towards the former, as the fitness function is based on the training data. Nonetheless, this positive correlation is rather intuitive. An active TTS will generally generate buy and sell signals at higher frequency within a trading period. Thus, it is most probable that such a TTS will consistently generate active schedule for both the training data and test data.

These observations seem to suggest that training and test returns are related indirectly via risk instead of the direct relationship that is widely assumed in single-objective approaches. Specifically,

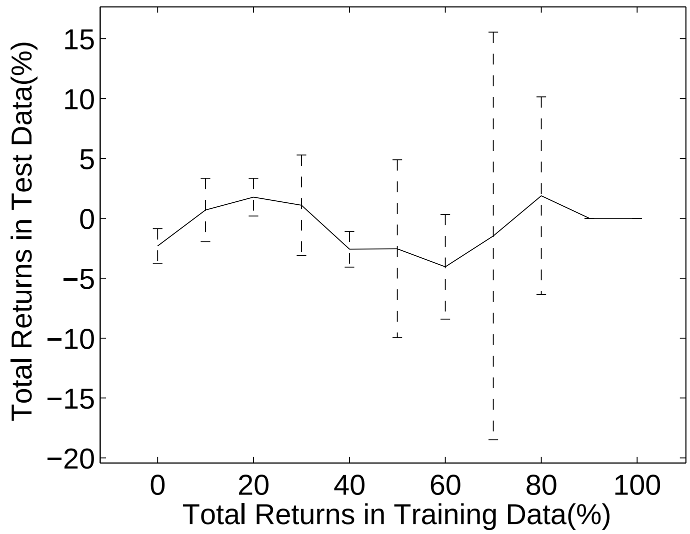


Figure 6.22: Statistical distribution of the average Test Returns at discrete values of Training Return of 10. The vertical lines denote the standard deviation of the weight at each value of risk exposure.

a TTS that yields high training returns is most likely to be associated with higher training risk by virtue of the risk-returns tradeoff. Due to the positive correlation between the training risk and test risk, this TTS will most likely trade actively in the test data. Consequently, since it is not optimized to the test data, this results in test returns of high volatility. The noisy version of the risk-return frontier in the plot of test returns and test risk could be attributed to these factors. Of course, these assertions are merely hypothetical and should be further verified by simulations. Nevertheless, it does show that positive correlation between training returns and test returns assumed in conventional single-objective approaches of ETTS optimization does not necessarily hold for all cases.

6.4.3 Generalization Performance

To formally evaluate the generalization performance of the ETTS found by the proposed MOEA, a total of ten different data sets were used to compensate the high dependence of generalization performance on the choice of the trading period. Each set comprised of one full year of test data, starting from the first trading day of January to the last trading day of December from the year 1995 to 2005, while the previous three years of trading data were used as the training set as shown in

Table 6.5. 100 days of historical data prior to the first trading day of the training set were included also.

Table 6.5: Generalization performance of MOEA over 10 different data sets.

Index	Training Set		Test set	
	Start	End	Start	End
96	04-03-1993	29-12-1995	02-01-1996	31-12-1996
97	03-01-1994	31-12-1996	03-01-1997	31-12-1997
98	03-01-1995	31-12-1997	02-01-1998	31-12-1998
99	02-01-1996	31-12-1998	04-01-1999	30-12-1999
00	03-01-1997	30-12-1999	03-01-2000	29-12-2000
01	02-01-1998	29-12-2000	02-01-2001	31-12-2001
02	04-01-1999	31-12-2001	02-01-2002	31-12-2002
03	03-01-2000	31-12-2002	02-01-2003	31-12-2003
04	02-01-2001	31-12-2003	02-01-2004	31-12-2004
05	02-01-2002	31-12-2004	03-01-2005	30-12-2005

Performance evaluation is not as straightforward in multi-objective approach as compared to single-objective approach. For the latter, the sole solution obtained at the end of the evolutionary process will be used to quantify the overall generalization performance. However as a set of solutions will be obtained instead in multi-objective optimization, this led to the selection problem of choosing the appropriate solutions for the evaluation of its generalization performance. An easy approach is to simply consider the average performance of the various ETTS obtained but this does not account for their varying degree of risk averseness.

Instead, ETTS obtained from the training set is classified according to the training risk in regular interval of 0.1 due to the positive correlation between training and test risk measure. They are then applied to the test data and the average returns in each group are calculated, as listed in Table 6.6. The returns for the buy-and-hold strategy are included also as a basis for comparison. It should be highlighted that the returns of the buy-and-hold strategy are extremely volatile, depending entirely on the trading period i.e. high returns could be reaped during bull markets where investor confidence is high, leading to widespread financial asset appreciation, and vice versa for bear markets.

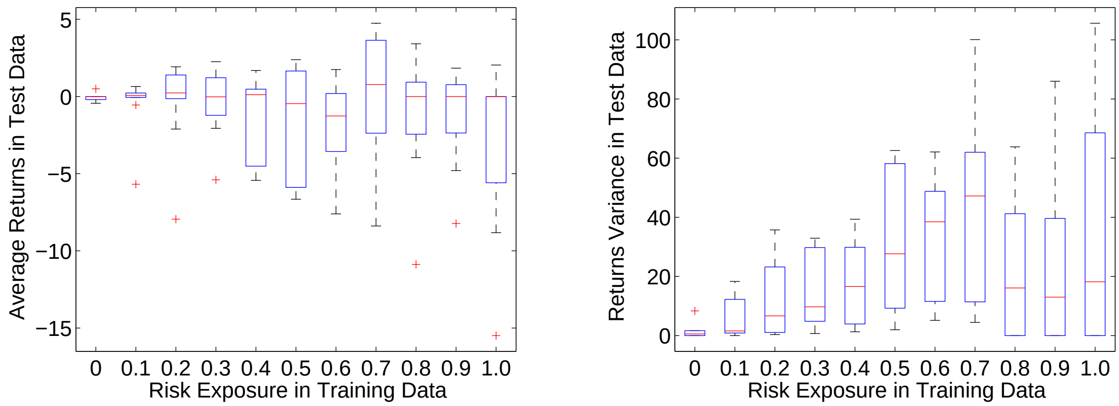
On the other hand, the returns for ETTS are much more conservative. In period where the buy-and-hold strategy can yield a profit of 76.60%, it can only achieve a maximum of 20.18% at a risk level of 0.3. And when the market dropped by 31.72% in 1997, the ETTS are able to generate positive returns for all risk level. These conservative results could be partly due to the averaging of solutions within the risk intervals.

Table 6.6: Generalization performance of MOEA over 10 different set of test data.

Index	Buy-and-hold Returns	MOEA Returns										
		Risk Level										
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
96	-2.33	0.16	-4.77	-7.48	11.52	6.52	8.09	0	0	0	0	0
97	-31.72	0.28	4.02	1.85	0	5.17	9.41	13.92	0	0	0	0
98	-8.76	0.38	16.75	-1.9	9.82	0	0	-27.18	0	0	0	0
99	76.6	-4.34	-2.18	-0.06	20.18	-7.76	0	0	12.35	9.15	0	0
0	-25.9	1.07	7.89	-0.67	5.11	9.28	-15.12	-8.3	-3.83	-5.57	0	0
1	-14.88	-6.79	-6.89	-12.45	-10.28	-0.07	0	0.31	-7.91	-8.81	0	0
2	-18.01	0.49	-0.24	-1.76	-4.16	-5.79	0	-22.58	-18.74	0	0	0
3	31.58	0.6	3.17	9.48	4.85	1.82	-7.06	-3.75	0	0	0	0
4	14.384	0.04	0.44	-0.61	1.53	0.92	0.96	-1.26	0	2.08	-2.85	0
5	12.89	0.04	-1.58	-4.08	-0.92	-3.79	-1.08	4.35	0	0.06	0	0
Arithmetic Sum	34.31	-8.06	16.61	-17.69	37.64	6.31	-4.79	-44.48	-18.12	-3.09	-2.85	0

Considering the arithmetic sum of the returns generated by the ETTS at the various risk level, its test returns can only outperform the buy-and-hold strategy at one particular risk level. However, it should be highlighted that the returns of the buy-and-hold strategy is attained at the maximum risk level as this strategy remain in the open position for 100% of the trading period.

To reinforce the claims in the previous section, Figure 6.23 plots the mean and variance of the test returns versus training risk for the various data sets. In Figure 6.23(a), the mean of the test returns fluctuates around the zero mark as training risk increases. Positive returns could hence be obtained if the transaction cost of 0.5% is not considered. This result is consistent to that obtained by Allen and Karjalainen [5] where their evolved rules did not generate excess returns over buy-and-hold strategy after the inclusion of transaction costs. The steady increase in the variation of the test returns in Figure 6.23(b) further verified the hypothesis that higher training returns correspond to increased volatility in the test returns. The drop after risk level of 0.8 is statistically insignificant due to the lack of solutions in that region.



(a) (b)

Figure 6.23: (a) Mean and (b) variance of the test returns for the data sets

Ideally, the multi-objective evolutionary platform could evolve a set of TTS with different level of risk averseness so as to suit the different types of investors, from conservative to risky. However, the erratic behaviors in the generalization performance, caused by the low correlation between the training returns and test returns, complicate the task of choosing the appropriate ETTS for practical implementation. Nevertheless, the simulation result does reveal some interesting insights to this problem, in particularly, the observation that the positive correlation between training returns and test returns assumed in conventional single-objective approaches of ETTS optimization does not necessarily hold for all cases.

6.5 Summary

In this chapter, a multi-objective evolutionary approach to the development of TTS was investigated here, where total returns and risk exposure were simultaneously optimized. Popular technical indicators used commonly in real-world practices were used as the building blocks for the trading agents, allowing the examination of their trading characteristics under an evolutionary platform. The Pareto front obtained by the algorithm accurately depicts the inherent tradeoff between risk and returns. The analysis of the TI composition along the risk-return frontier reveals that each TI has varying degrees of significance in different regions of the tradeoff surface depending on their underlying characteristic. As such, future work will certainly involve profiling other TI to further understand their trading characteristics.

The correlation study suggested that the returns from the test and training data are not correlated in this context, which is contrary to popular belief in the single-objective approach of this optimization problem. Instead, higher returns in training data only corresponds to larger volatility in the returns generated in the test data. This is further reinforced by the erratic trends in the analysis of the generalization performance. Nevertheless, in order to further validate the evolutionary model and the empirical results, it is absolutely necessary to subject the evolutionary model to other data sets and include different TI that are able to detect other market signal not covered by the current group of TI.

Chapter 7

Dynamic Index Tracking via Multi-Objective Evolutionary Optimization

7.1 Introduction

Markets indices - such as the Dow Jones Industrial Average, the NYSE Composite and the Standard & Poor's 500 - track the performance of a customized basket of securities that best represent the underlying financial market/sector. These indices provide an indication of the overall market performance, aiding investors in their investment decision. They function also as benchmarks against which investors can evaluate the performance of their portfolios. There has been an accelerating trend in recent decades to create passively managed portfolios that are based on market indices, known as index funds i.e. an investment vehicle which exhibits only market risk, with all other risks having been diversified away.

Essentially, index tracking denotes a passive investment strategy targeted to replicate the performance of a specific financial market index, regardless of market conditions. In recent years, passive portfolio management has been gaining in popularity as empirical studies have revealed that sustainable and stable yields exceeding market average are rare. In fact, advocates claimed that passive strategies routinely beat a large majority of actively managed mutual funds after cost [86, 160, 223]. Since index funds attempt to replicate the holdings of an index, they obviate the need forand thus many costs ofthe research entailed in active management, leading to a lower "churn" rate (the turnover of securities which lose fund managers' favor and are sold, with the attendant cost of commissions and capital gains taxes).

The direct approach for index tracking is full replication where the index fund is constructed by weighing the index constituents as per their exact proportion in the underlying index. Though simple conceptually, it poses significant challenges in real-world implementation. Firstly, certain index constituents might be held in small quantities which are inconvenient and costly to administer. Furthermore, the net asset value of the index fund might not be substantial for full index replication. In lieu of these limitations, partial replication represents a plausible alternative, which considered only a subset of securities from the index. Nevertheless, the index replication process is complicated also by random capital flows within the index fund during the investment horizon and dynamically changing index arising simply from market price movements and/or structural revisions like merger, stock splits, etc. To adapt to the ever-changing market conditions, the tracker portfolio has to adjust continually during the investment horizon, otherwise known as rebalancing. As transaction costs associated with the purchase and sale of securities are inevitable, index replication should provide means of limiting these costs from scaling up, besides focusing on tracking performance. However, the reduction of transaction and management costs will typically be at the expense of wider returns deviations between the tracker funds and the underlying index by virtue of their inherent trade-offs.

Constructing an index fund via partial replication, which involves choosing the appropriate subset of securities constituents and their corresponding proportions, is non-trivial, especially when the underlying index comprise of many securities as empirical studies have revealed that a minimal number of securities are required for a basic level of tracking performance [188]. While many optimization tools and heuristic techniques have been considered in related literature, evolutionary computation, a class of stochastic search technique has been gaining significant attention due to its success in its capability in dealing with highly complicated search space. Moreover, they can be extended to multi-objective optimization in handling problems with competing specifications, as their population-based search approach enables them to sample a pool of solutions simultaneously during the optimization process.

This chapter aims to leverage the search capability of evolutionary algorithm and construct an index tracker system targeted to reduce tracking error and transaction costs simultaneously during the investment horizon. Furthermore, a multi-period variant of the index tracking optimization problem will be considered, which requires rebalancing throughout the investment horizon due to changing market conditions and stochastic capital injections/withdrawals from the investors. Specifically, the proposed multi-objective evolutionary index tracking optimizer (MOEITO) will be invoked continually during the investment horizon to align the tracker portfolio to the dynamic market conditions. Lastly, the inherent flexibility of the evolutionary algorithmic platform allows realistic constraints like lot constraints, budget constraints and tier-based transaction costs to be considered.

The remainder of this chapter is structured as such. A formal definition of the index tracking optimization will follow, covering the general algorithmic flows and the specific objective functions and constraints. Following that, a brief overview of evolutionary optimizers will be presented, together with a discussion on how it can be adapted for the purpose of index tracking. Simulation results and analysis to validate the proposed MOEITO will follow suit before the conclusions are drawn.

7.2 Index Tracking

Index tracking denotes the optimization problem of constructing a portfolio that can replicate the performance of a financial index, independent of market conditions. While many indices are available for various asset markets, for example the Standard and Poors 500 Index (S&P500) for US equities, the Goldman Sachs Commodity Index for commodities, the Lehman Brothers Bond Index for bonds, etc. [29], this chapter deals exclusively with equity index funds i.e. a portfolio of equities chosen to reproduce the returns characteristics of a specific equity index.

7.2.1 Variable Notations

Tables 7.1 and 7.2 list the notations for the various variables in the index tracking optimization problem.

Table 7.1: Variable notations for the index tracking optimization problem.

Variable Notation	Description
N	Total number of distinct securities that can be included in the tracker portfolio. While the set of securities typically mirrors the index constituents, off-index securities that can proxy the underlying constituents can be considered also.
$T = \{0, 1, ..., T\}$	The investment horizon is split into T time periods, where each time period, t is potentially associated with a portfolio decision. The time period can be daily, weekly, quarterly, etc depending on the rebalancing frequency. Weekly time period is considered here as the simulation data sets comprised of weekly price data
$Q_t = \{q_{t,1}, q_{t,2}, ..., q_{t,N}\}$	$q_{t,i}$ represents the quantity of stock i ($i = 1, 2, ..., N$ held in the tracker portfolio at time t. Collectively, the vector, Q_t defines the security composition of the portfolio. Q_t will be updated only (if necessary) at the beginning of the time period, t and remained constant thereafter until $t + 1$ (See Figure 7.1).
$P_t = \{p_{t,1}, p_{t,2}, ..., p_{t,N}\}$	$p_{t,i}$ represents the closing price of stock i at time t. The price vector P_t will be updated at the end of the time period, t and remained constant thereafter until the end of $t + 1$(See Figure 7.1).
I_t	The market value of the index at the end of time period, t.
R_t	The index returns at the end of time period, t. Without any loss in generality, geometric returns is considered here (i.e. $R_t = \frac{I_t - I_{t-1}}{I_{t-1}}$), though logarithmic returns are widely considered also.
V_t	The aggregate market value of the securities in the tracker portfolio at time t. Value here could either be based on beginning of the time period, $BV_t = \sum_{i=1}^{N} p_{t-1,i} q_{t,i}$ or end of the time period, $EV_t = \sum_{i=1}^{N} p_{t,i} q_{t,i}$. The calculation difference in calculation is attributed to P_t being the end-of-period price. Hence, calculation of BV_t considered the previous periods price i.e. P_{t-1}. V_t in this chapter will referred to EV_t by default, as portfolio decisions will be implemented at the end of each time period based on EV_t.
r_t	The returns of the tracker portfolio for time period t, calculated by $r_t = \frac{EV_t - BV_t}{BV_t}$. will remain static over t and any returns are primarily attributed to price movements of underlying portfolio constituents.

Table 7.2: Variable notations for the index tracking optimization problem (contd).

Variable Notation	Description
C_t	The cash position in the tracking portfolio. Due to the finite divisibility of securities, cash positions in the portfolio are inevitable. $C_t > 0$ and $C_t < 0$ represents long and short cash position respectively, which can be modeled to earn/pay short-term cash rates i.e. Libor, REPO and etc.
ΔC_t	The change in cash position for time period t, which could be due to interests due to cash positions, capital injection/withdrawal etc.
TC_t	Decision on portfolio adjustments (if necessary) will be made and implemented at the end of each time period and the corresponding transactions will be based on the closing price. Consequently, the transaction cost will be a function of the closing price at time period t, the quantity composition held at the beginning of the time period Q_t and the new quantity composition Q_{t+1}, which will take effect at the beginning of the next time period $t+1$ i.e. $TC = f(Q_t, Q_{t+1}, P_t)$
F_t	The total value (end period) of the tracker portfolio at time t, which mainly includes the market value of the securities and the cash at hand net of any transaction costs incurred at time t, $F_t = V_t + C_t - TC_t$

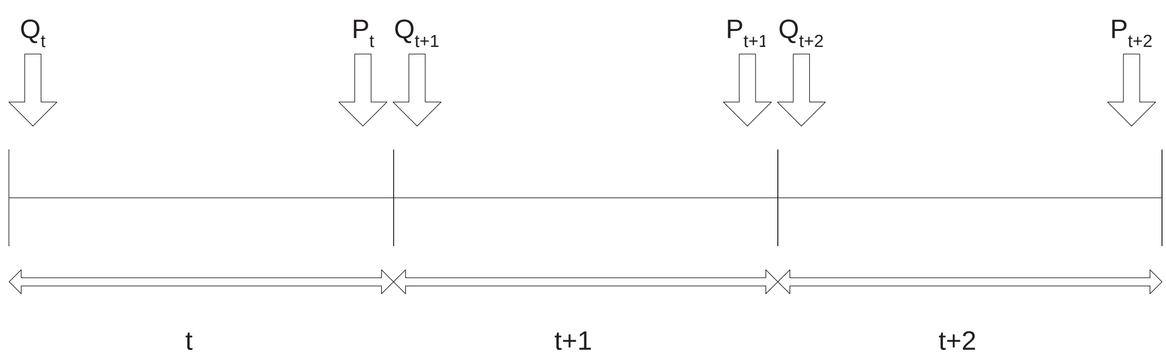


Figure 7.1: Chronological sequence in which equity prices and portfolio quantity are updated. Prices will be updated at the end of each time period based on closing market prices. All transaction is assumed to be executed at the end of the time period based on the updated prices and the new quantity composition will be reflected at the beginning of the next time period.

7.2.2 Problem Definition

Mutual funds are collective investment schemes that pool money from investors at large and professionally managed in accordance to pre-specified investment mandates for example aggressive growth, capital appreciation, tax-free bond and etc. They can be broadly classified as closed and opened, where the invested capital for the former remained fixed throughout the investment horizon and portfolio management for the latter is complicated by capital injections/withdrawals at the discretion of the investors, hence garnering closer monitoring and management of the funds.

The index tracking optimization problem is analogous to a mutual fund that focuses on index replication. Based on a fixed capital allocated prior to the investment horizon, an initial portfolio that could possibly track the underlying index will be built. In this context, the constituent composition will be determined by MOEITO. Due to the finite divisibility of securities, cash positions in the portfolio will be inevitable. If the cash position is positive/negative, the fund will earn/pay a pre-specified money rate, for example repo or Libor rate. The existence of a cash position during the investment horizon will result in a performance mismatch between the tracker portfolio and the underlying index arising from differences in index movement versus the money rates. As such, it is imperative to minimize the cash holdings during this period.

During the investment horizon, returns deviations between the tracker portfolio and the underlying index are inevitable due to the manifestation of a static tracker portfolio against the dynamic market conditions. As such, the tracker portfolio has to continually change its constituent composition and weighting to overcome serious performance deviation [171]. Typically, frequent portfolio rebalancing will enable the incorporation of more timely information, ultimately leading to better tracking performance [89].

Portfolio rebalancing can be triggered at fixed frequency or based on pre-specified criteria. For the former, portfolio rebalancing will be executed in regular intervals regardless of any market changes. As the rebalancing frequency will remain static over the investment horizon, inappropriate frequency will result in excessive transaction costs and/or significant deviations from the desired tracking performance. Typically, higher rebalancing frequency will lead to better tracking performance but translating to more transactions and hence higher transaction costs [168]. Event-triggered rebalancing represents a more flexible alternative, where the defined criteria depend on the investors goals and preferences. Miao [171] considered market volatility since heightened volatility will result in structural changes within market indices. Meade and Salkin [168] considered portfolio-specific criteria, namely ex-post tracking error and returns deviations exceeding certain pre-defined limits. In this context, random capital injection/withdrawals within the investment horizon will also trigger rebalancing, as the excess cash position from injection has to be allocated to relevant equity positions or existing holdings have to be liquidated to free up cash for redemption. Capital injection/withdrawals will be broadly referred as injections as withdrawals are simply negative injections.

Rebalancing will occur (if necessary) at the end of each time period, t and the tracker portfolio will be re-configured with the available capital being the market value of the existing holding V_t, inclusive of the cash position, C_t. MOEITO will be invoked to re-align the tracker portfolio to the market conditions at the end of time period, t. After the new tracker portfolio composition has been determined, transactions will be executed based on P_t, ignoring transaction timing effects, and Q_t will be updated accordingly. Immediate exposure to index constituents is assumed here, which is viable in real world via derivative instruments that allow investors to get immediate exposure to the underlying before converting them to equities at later period.

Equation 7.1 describes the temporal relationships between the associated variables during rebalancing, where the consistency of the portfolio value before and after the transactions are maintained. Essentially, the aggregate market value of the tracker portfolio and the cash position before and after the transactions should be equal net of the transaction cost and cash changes. In other words, the cash position at $t + 1$ will be the cash held at t adjusted for changes in portfolio values due to the transactions, increment in cash position and the corresponding transaction cost (7.2).

$$C_{t+1} + P_t \cdot Q_t = C_t + P_t \cdot Q_{t+1} + \Delta C_t - TC_t \tag{7.1}$$

$$C_{t+1} = C_t + P_t \cdot Q_{t+1} - P_t \cdot Q_t + \Delta C_t - TC_t \tag{7.2}$$

At the end of the investment horizon, the entire tracker portfolio will be liquidated and the performance of the tracker fund over the investment horizon can be quantified.

7.2.3 Objective Functions

Although the ultimate objective is to build a tracker portfolio with return characteristics identical to the underlying index, a range of objective functions have been considered in related literature. Tracking error, a statistical measure that quantifies the returns differences between the tracker portfolio and the underlying index, is the widely considered measure [12, 55, 118, 168, 215]. Alternatively, the returns deviations have been viewed from a risk management perspective [89], where risk measures like Value-at-risk [90, 200] and downside variance/deviation [167] have been implemented as objective functions. Besides measures specific to returns deviation, supplementary criteria like transaction costs and excess returns over the index have been considered also [12, 168, 230]. The former improves the operational aspect of portfolio management while the latter arises from enhanced indexation strategy [198]. Nevertheless, most literature works so far have considered index tracking as a single-objective optimization problem, despite the wide range of inherently conflicting objective functions. The proposed MOEITO will formulate index tracking as a multi-objective problem, where tracking error and transaction cost will be simultaneously minimized during the optimization process.

Tracking Error

Tracking error typically centers on the returns deviation between the tracker portfolio and the underlying index. Widely considered measures in related works are the mean square error (7.3) and tracking error variance (7.4) with the over-bars denoting average value. While MSE directly measures the cumulative returns differences, TEV measures the deviation of the returns differences which in case of large portfolio deviations will not be adequately quantified by MSE.

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(r_i - R_i)^2 \tag{7.3}$$

$$TEV = \frac{1}{T}\sum_{t=1}^{T}\left[(r_i - R_i) - (\bar{r_i} - \bar{R_i})\right]^2 \tag{7.4}$$

While the ultimate objective is to optimize a tracker portfolio that could follow the index closely during the investment horizon, the challenge is that the only information available during portfolio build-up i.e. $t = 0$ are historical price data prior to the investment horizon. Inevitably, the tracking performance during the optimization process, which is based on historical information, will definitely be different from its actual performance in the investment horizon. Nevertheless, such methodology is still widely adopted in almost all related works, as the underlying assumption is that the historic data contain useful information about the future and consequently, portfolio with desirable past behavior will replicate similar characteristics in the unseen future. It should be highlighted that the desired behavior here specifically pertains to the behavior relative to the index and not to the index itself. Clearly, the applicability of such relationships in the future is highly doubtful as external shocks to the industry can easily disrupt them in the short term or lead to structural changes. The highly debatable basis of this assumption motivates the use of ex-ante tracking error, which exhibits better predictive capability as compared to ex-post tracking error.

Generally, if tracking error is measured historically, it is called realized or ex-post tracking error; if a model is used to predict tracking error, it is called ex-ante tracking error. The former is more useful for reporting performance; whereas the latter is basically a predicted TE based on a certain risk model and is generally used by portfolio managers to control risk. One main difference is that portfolio composition, which will be stochastic ex-post, is assumed to be stochastic ex-ante [110]. While various types of ex-ante tracking error models exist, from simple equity models which use beta as a primary determinant to more complicated multi-factor fixed income models, it will simply be the MSE (7.3) calculated based on the quantity composition implied from the chromosomal representation of the tracker portfolio subjected to historical returns series.

Transaction Costs

Transaction costs associated with the purchase and sales of securities are inevitable in real-world. For brevity, discussions on transaction costs are omitted here as they were already covered in Chapter 5. Transaction costs will be directly considered as a objective function here and the transaction cost function will follow (5.10). The importance of transaction cost in index tracking actually lies during rebalancing where the portfolio constituents have to realign to the market conditions, as they can be especially significant for index with many underlying constituents for example, Russell 3000 index and Wilshire 5000 index or during events of radical changes within the index structure. The level of transaction cost during rebalancing in the proposed index tracking system is controlled by the simultaneous optimization of both tracking error and transaction cost, generating a portfolio with adequate tracking capability yet relatively similar to the existing holdings, translating to lower transaction cost. As transaction cost can easily scale up with higher rebalancing frequency, especially in volatile market conditions, rebalancing instances are moderated in the index tracking system where it will be triggered only when ex-ante tracking error exceed the pre-defined limit.

7.2.4 Constraints

In real-world implementation, the quantity composition of the portfolio will be subjected to the round-lot and the non-negativity constraints, mathematically formulated as follows:

$$q_{i,t} \mod l_i = 0 \tag{7.5}$$

$$\text{s.t. } q_{i,t} \leq 0$$

where l_i denotes the trading lots for asset i. As mentioned in Chapter 5, the round-lot constraint requires the number of any asset included in the portfolio to be in exact multiples of the normal trading lots [152, 162, 221]. The inclusion of round-lot constraint will relax the conventional budget constraint, which will consequently generate excess cash positions in the tracker portfolio during the investment horizon as denoted by C_t. The non-negativity constraint is defined in (7.5) also, where only long positions are allowed in the tracker portfolio. While this can be removed for generality, the non-negativity constraint is included here as index constituents typically assumed long positions in the underlying index.

7.3 Multi-Objective Evolutionary Optimization

Many different approaches have been proposed for index tracking, ranging broadly from mathematical/statistical methods like quadratic programming [168], constrained aggregation technique [182] and factor model [55] to heuristic models like fuzzy logic [80], neural networks [264] time series clustering [87], hierarchical clustering [66], threshold accepting algorithm [93] and simulated annealing [142]. Interested readers are referred to [12] for a further discussion of related literature works on index tracking. Recent developments predominantly relied on heuristic approaches, as they generally allow flexible problem formulation and avoid cumbersome computational efforts. As highlighted earlier, the optimization platform considered in this chapter will be based on multi-objective evolutionary algorithms, a class of heuristic technique that has been gaining significant attention for its capability in dealing with highly complicated search space and multiple objectives throughout the optimization process.

Based on basic concepts from the biological model of evolution, the search dynamic of evolutionary algorithms are driven by biologically inspired evolutionary operators like selection, crossover and mutation, which will explore and exploit the associated search space for the optimal solution. The crossover and mutation operator manipulate and create potential solutions, while the selection operator provides the necessary convergence pressure. The strength of evolutionary algorithms lies in their population-based search approach, which will generate higher diversity in the search space, reducing the likelihood to converge to the local optimum, suitable for tackling high dimensional search space problems like the index tracking problem [12, 117, 118, 181, 188, 215].

7.3.1 Chromosomal Representation

Evolutionary algorithms maintain a population of chromosome during the optimization process, where each of them represents a potential solution, which in the context here denotes a portfolio of equities. Hybrid representation will be considered here with the fitness evaluation function modified to handle the lot constraint. Consider an index with only 4 underlying constituents, an instance of the hybrid representation (highlighted in grey) is shown in Figure 7.2. Essentially, the binary vector denotes the inclusion of securities in the portfolio i.e. $b_i = 1$ means that the i-th security is included, vice versa. Correspondingly, the weight vector denotes the proportion of capital that is allocated to each security, which is typically normalized to adhere to the budget constraint.

In the absence of the round-lot constraint, the total capital will be fully expensed and distributed to the various securities in accordance to the weight vector. As \$25 is allocated to the 3rd security and its market price is \$2, this results in 12.5 lots of the 3rd security being bought, violating the round-lot constraint. Conventionally, this can be rectified by rounding the quantity to the nearest integer, though this will result in a mismatch between the portfolio value and capital i.e. rounding

	Equities No.	**1**	**2**	**3**	**4**	
	Market Price per Security	$1	$3	$2	$5	
	Binary Vector	1	0	1	1	
	Weight Vector	0.25	0	0.25	0.50	
Without Round-lot Constraint	**Quantity**	25	0	12.5	10	
	Value Allocated	$25	$0	$25	$50	$100
Round-up	**Quantity**	25	0	13	10	
	Value	$25	$0	$26	$50	$101
Round-down	**Quantity**	25	0	12	10	
	Value	$25	$0	$24	$50	$99

Figure 7.2: Illustration the fitness evaluation function in handling the lot constraint.

up (down) the quantity will result in the portfolio value being more (less) than the initial budget, resulting in the portfolio being overweight (underweight) in securities and underweight (overweight) in cash. This excess cash position will result in performance mismatch between the tracker portfolio and the index, depending on the index movements.

To minimize the cash position during the investment horizon whilst maintaining a neutral view on the index movements, each security quantity will be randomly rounded up or down to the nearest integer during the fitness evaluation process. As this rounding procedure is an iterative process, the probability of security quantity being rounded up or down will adjust whether tracker portfolio is overweight or underweight with respect to the expensed budget. While views on index movements can be incorporated via changing the proportion of securities quantity being rounded up/down, this is not considered here as it will escalate tracking error and excess returns over the index is secondary in this chapter.

7.3.2 Selection Process

Index tracking is formulated as a multi-objective problem in this chapter, which involves the minimization of the ex-ante TE and the transaction costs. In contrast to single-objective optimization, the optimal solutions for a multi-objective optimization problem exist in the form of alternate trade-offs known as the Pareto-optimal set. Each objective component of any non-dominated solutions in the Pareto-optimal set can only be improved by degrading at least one of its other objective components. The Pareto-optimal set when plotted will illustrate the trade-off between the various objectives, as illustrated in Figure 7.3.

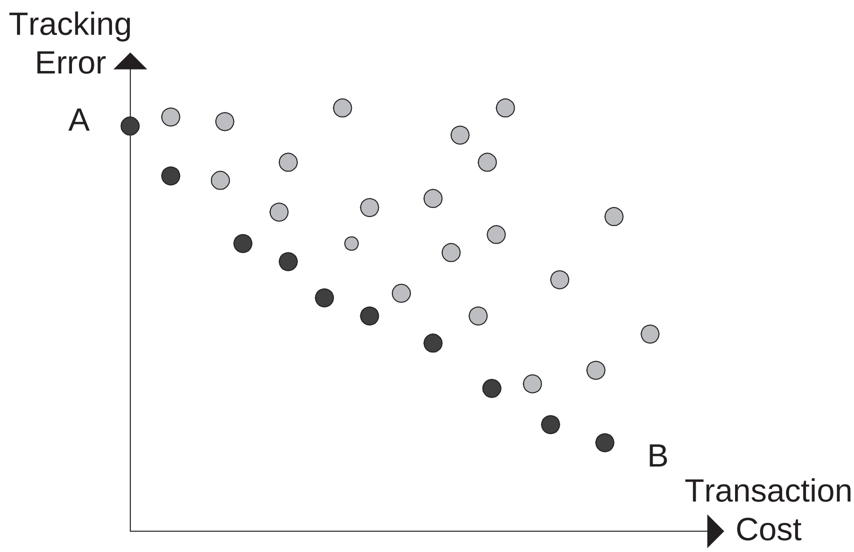


Figure 7.3: Illustration of the tradeoff between tracking error and transaction cost.

Each point denotes a potential tracker portfolio and the black and gray circles represent non-dominated and dominated solutions respectively, where the former denotes the Pareto optimal set as their tracking error cannot be minimized further without compromising transaction cost. In the context of single-objective optimization where tracking error is the sole priority, the evolutionary process will ultimately drive the solutions towards the extreme point B, which might not be economical sustainable for index funds in general, as they aim to charge the lowest management fees possible. Point A represents the extreme case of leaving the tracker portfolio unchanged during the rebalancing process, resulting in zero transaction cost.

Unlike conventional single-objective approach which will generate a single unique solution at the end of the optimization process, the multi-objective approach will generate a set of Pareto optimal solutions instead. In the total absence of information regarding the preference of objectives, this naturally led to the question of which tracker portfolio should be selected eventually to track the index. Ideally, the selected tracker portfolio should strike a balance between TE and transaction cost. The proposed selection methodology is outlined in Figure 7.4. A reference point C is created based on the minimal TE and transaction cost of the extreme solutions i.e. points A and B. The Euclidean distance between each solution and the reference point C is calculated and the solution closest to C be selected as the tracker portfolio. To prevent any disparity between the two objectives, each objective value is normalized within their range before calculation of the Euclidean distance.

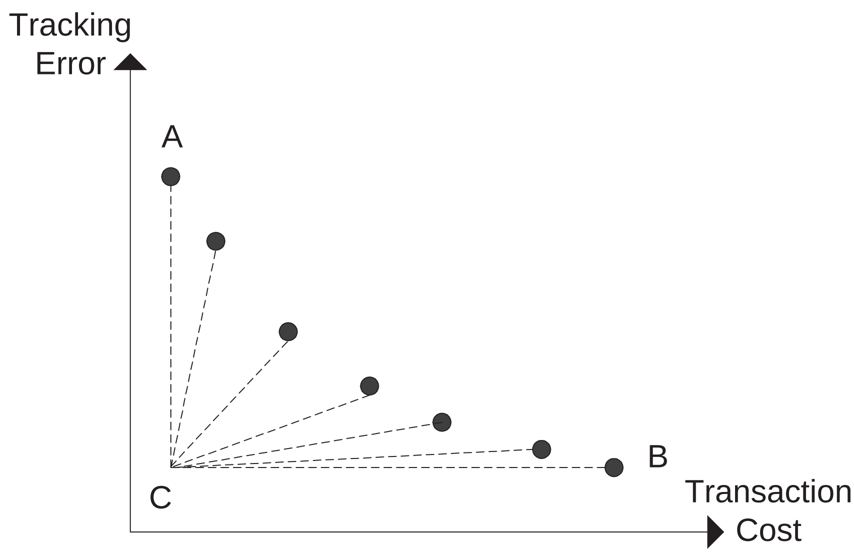


Figure 7.4: Illustration of the selection process for the tracker portfolio.

7.3.3 Dynamic Archiving Operator

Rebalancing the tracker portfolio to the ever-changing market conditions is akin to dynamic optimization where the problem underlying specifications e.g. objective functions, constraints and etc, vary with time, consequently altering the optimal solution. The most direct approach is to treat each change as a separate problem and re-initiate the optimization process, whenever changes in the problem environment are detected [194]. However, the re-initialization of the evolving population constitutes a loss of genetic materials from the previous optimization, which might otherwise be useful if the environment changes are not drastic. Furthermore, this explicit restart approach assumes that any changes in the environment could be identified, which unfortunately might not always be the case [120].

To preserve information between rebalancing occurrences throughout the investment horizon, an external archive will be maintained, where the archiving strategy aims to simultaneously maximize genetic diversity (in the decision space) and minimize proximity with the current optimal solution (in the objective landscape). The former is indirectly measured by the transaction costs in switching the portfolio constituents, while the latter can be quantified by the ex-ante TE as well. While multi-objective formulation is conventionally being utilized to address the inherent trade-offs between several objectives in optimization problems, multi-objective formulation is employed here as an archiving strategy to sustain a population pool that can maintain a tradeoff between the two stipulated archiving objectives i.e. optimality and diversity.

Besides deciding on types of individuals to be stored, an archiving strategy should comprise of replacement and retrieval strategies. The former will simply be crowding measure based on niche count and solutions with higher niche count will be removed. Retrieval strategies are concerned with how the archived solutions are being utilized during the evolutionary progress. Retrieval will occur at the beginning of each portfolio rebalancing, where the archived solutions will be merged with a population of random generated solutions and the survivors being selected based on tournament selection. The archive will be updated at the optimization process where all solution will be re-evaluated based on the selected tracker portfolio.

7.3.4 Algorithmic Flow of Index Tracking System

The algorithmic flow of the multi-objective evolutionary platform is shown in Figure 7.5. At the start of the algorithm, the price series of the index and its underlying constituents are loaded to allow direct access of this information during algorithmic runtime, hence speeding up the computation time required.

The algorithm maintains a fixed size population throughout the evolutionary process. During the initialization process, tracker portfolios will be randomized until the population is filled. Specifically, it will involve randomly generating a set of binary and real vector within the cardinal range of the index. A fixed size archive is used to store the non-dominated solutions discovered during the evolutionary process. The archive is updated every generation, where agents that are not dominated by any members in the archive will be added into the archive and any members in the archive that are dominated by this new agent will be removed. The archive helps to ensure convergence by preventing the loss of good solutions due to the stochastic nature of the evolutionary process. In the selection process, elitism is implemented by selecting individuals to a mating pool through a binary tournament selection of the combined archive and evolving population. The selection criterion is based on Pareto ranking and in the event of a tie, the niche count will be employed. The mechanism of niche sharing is used in the tournament selection as well as diversity maintenance in the archive.

After one complete generation, the evolutionary process will repeat until a predefined number of generations are reached.

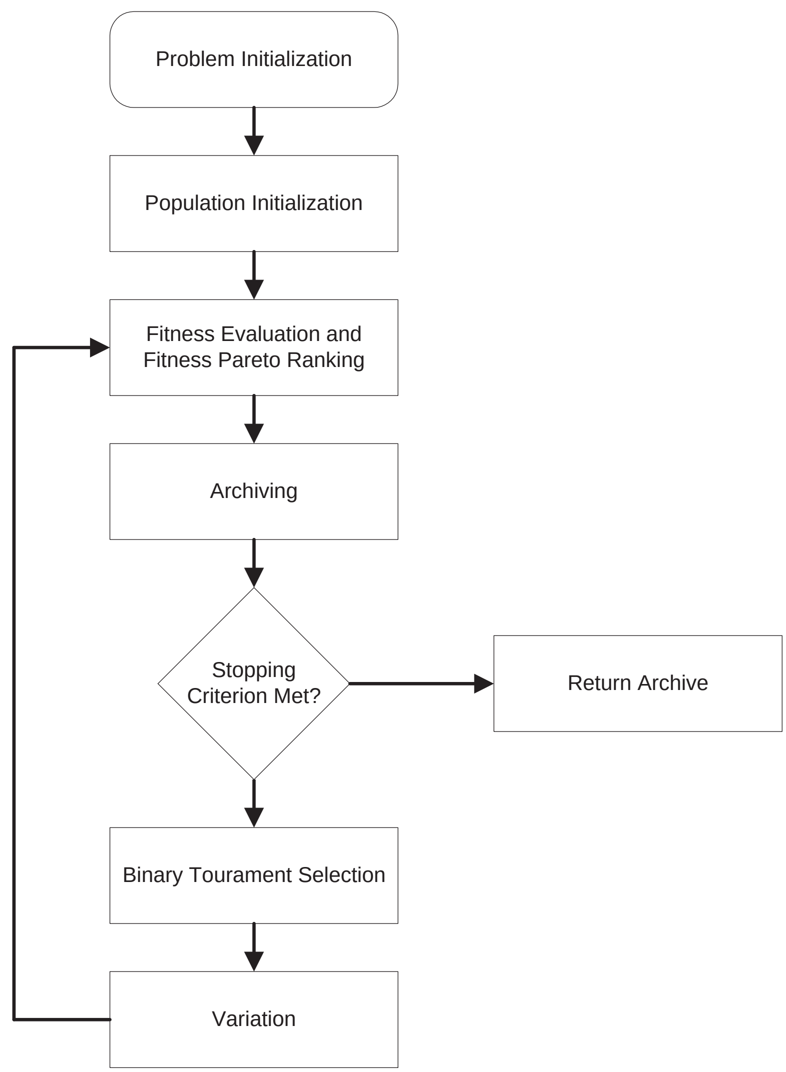


Figure 7.5: Algorithmic flow of the multi-objective evolutionary platform.

Table 7.3: Description of Simulation Data Sets.

Problem Index	Data Source	Number of Constituent
PORT1	Hong Kong, Hang Seng	31
PORT2	German, DAX 100	85
PORT3	British FTSE 100	89
PORT4	U.S. S&P 100	98
PORT5	Japanese Nikkei 225	225

7.4 Single-Period Index Tracking

This section will present the empirical results to validate the proposed MOEITO in a single-period instantiation of the index tracking optimization problem i.e. portfolio rebalancing will not be considered.

7.4.1 Data Sets & Simulation Setting

Five data sets obtained from the OR-library [11] were considered here. Specifically, they comprised of 291 weekly price data for five equity indices (and their underlying constituents) from major financial markets. Table 7.3 provides a brief summary of the data sets. While the stock universe of PORT2, PORT3 and PORT4 averaged around 90 assets, PORT1 and PORT5 have significantly lesser/more assets. Typically, greater computational efforts will be demanded to track an index with more underlying constituents due to a larger search space.

Portfolio composition will typically be based on the historical returns characteristics of the underlying constituents, where past returns profiles are assumed to persist in the future. The usual evaluation methodology for such predictive/forecasting model will be to split the data set into training set [1,145], where the model is optimized and test set [146,291], where the performance of the resulting model is evaluated eventually. Specially, the data set will be split in the proportion [1,145] and [146,291] respectively. Ideally, the model will attune to the training data via the evolutionary optimization process and its performance on the test data will approximate its generalization capability in real-world implementation.

Table 7.4: Algorithmic parameter settings of MOEITO for the simulations.

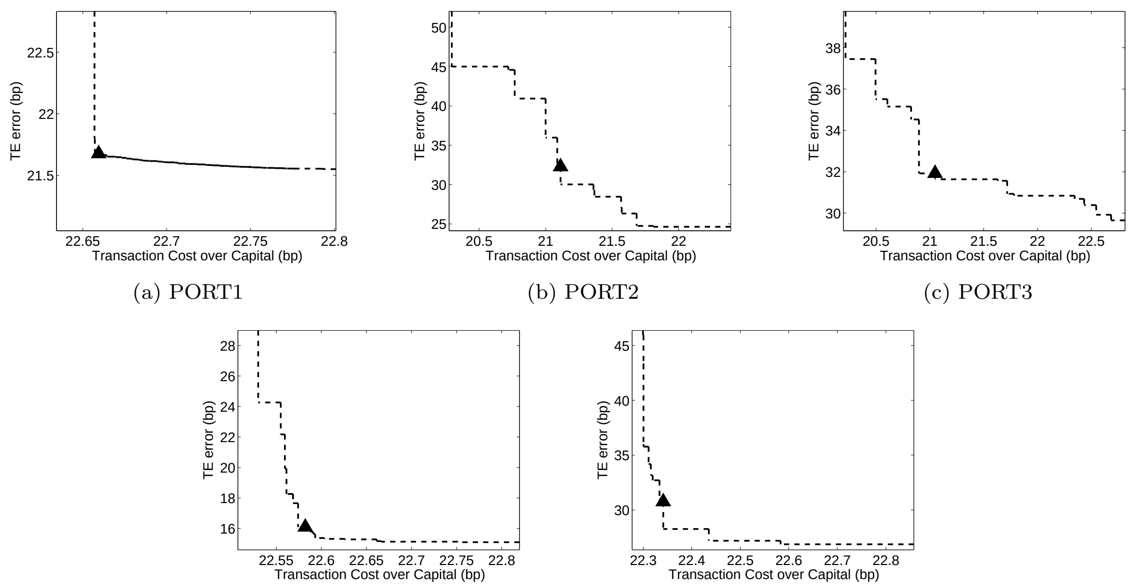
Evolutionary Optimizer	Total Generations	500
	Population Size	100
	Archive Size	100
	Selection	Binary tournament selection
	Crossover	Uniform crossover with probability 0.9
	Mutation	Bit-wise mutation with probability 0.01
Index Tracking	Initial Capital	100,000,000
	Lot Size	1,000
General	Simulation Runs	30

The greatest challenge for performance evaluation here lies in the fact that no optimal solution sets are available for the attained solutions to benchmark against, though conceptually the "optimal" solution should be zero ex-post TE and transaction costs, which is clearly unrealistic and unattainable in real-world implementation. As such, this chapter will assume that MOEITO can provide a reasonable approximation to the problems by virtue of the works in previous chapters. As such, the simulations will focus on operational aspects of the index tracking system and the sanity of the performance variation with respect to changes in the problem settings.

Unless otherwise stated, the parameter configuration of MOEITO in the simulations, including those in the subsequent sections, are listed in Table 7.4. In all simulations, 30 independent runs were made. Also for a more accurate and fairer comparison, the same random seed was assigned to each set of runs to ensure that they started with the same initial population.

7.4.2 Simulation Result & Discussion

Based on the solutions generated by MOEITO over 30 runs in the training sets, the attainable surfaces (Figure 7.6) clearly illustrate the tradeoff between ex-ante TE and transaction costs. While the variation range for ex-ante TE differed across problems i.e. 25-50bp in PORT2 versus 21.5-22.5bp in PORT1, transaction costs averaged between 20-23bp of the initial capital, corresponding to the average rates in the transaction cost schedule. Even though the transaction cost might seem relatively insignificant here, they can scale up considerably later in multi-period rebalancing, especially when the optimized portfolio differs significantly from the existing holdings.



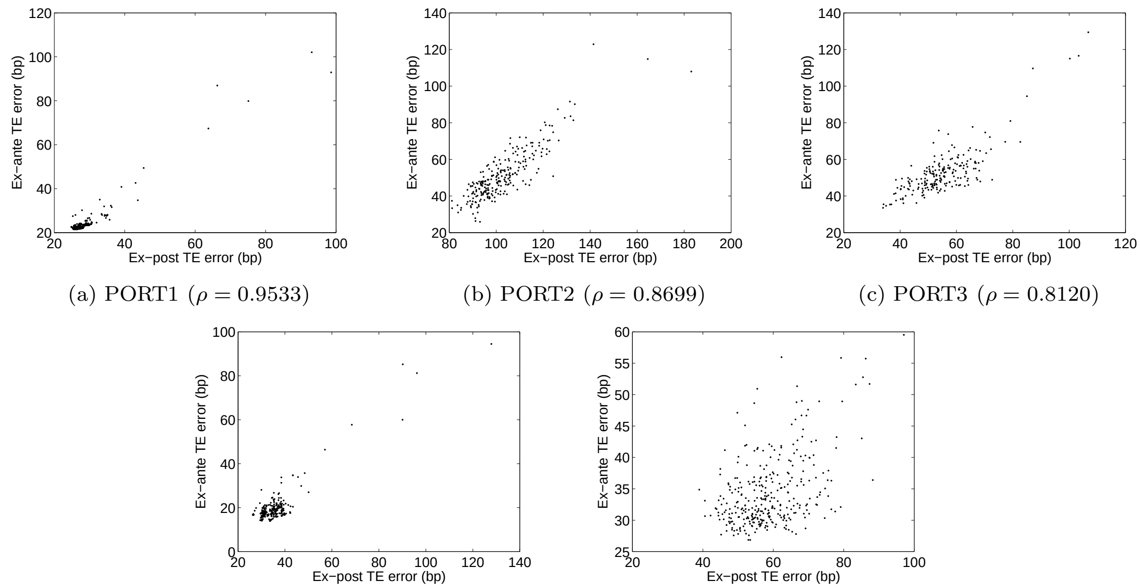
(a) PORT1 (b) PORT2 (c) PORT3

(d) PORT4 (e) PORT5

Figure 7.6: Attainable surface of ex-ante TE against transaction costs (bp, over initial capital) for the various training sets (over 30 algorithmic runs).

Ideally, the ex-ante TE based on the training data should closely approximate the ex-post TE attained in the test data, reflecting the generalization capability of the model. Their correlation over the 30 runs were calculated and shown in Figure 7.7, together with the corresponding scatter-plots. In general, a positive correlation was observed between the ex-ante and ex-post TE, though the degree varied widely across the data sets. While high correlations of above 0.9 were attained in PORT1 (0.9533) and PORT4 (0.9048), implying that lowering ex-ante TE will most likely result in better tracking performance in these test data, PORT5 attained a significantly lower correlation of 0.4742.

To further evaluate the generalization capability of MOEITO, the ratio of the ex-post TE over the ex-ante TE for each solution in the various data sets were calculated and their distribution is summarized in Table 7.5. Higher correlation between ex-ante and ex-post TE in PORT1 and PORT3 translated to a smaller ratio, illustrating the generalization efficacy of ex-ante TE in these



(a) PORT1 ($\rho = 0.9533$) (b) PORT2 ($\rho = 0.8699$) (c) PORT3 ($\rho = 0.8120$)

(d) PORT4 ($\rho = 0.9048$) (e) PORT5 ($\rho = 0.4742$)

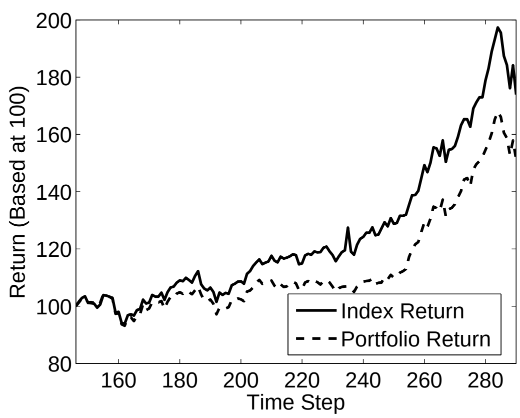
Figure 7.7: Scatter-plots of ex-ante tracking error versus the ex-post tracking error for the various solutions attained.

two data sets, which will consequently translate to lesser rebalancing demand and transaction costs when extended to multi-period index tracking. Conversely, ex-ante TE will underestimate the actual tracking performance for the rest of the data sets (by a factor of 2 for PORT2!).

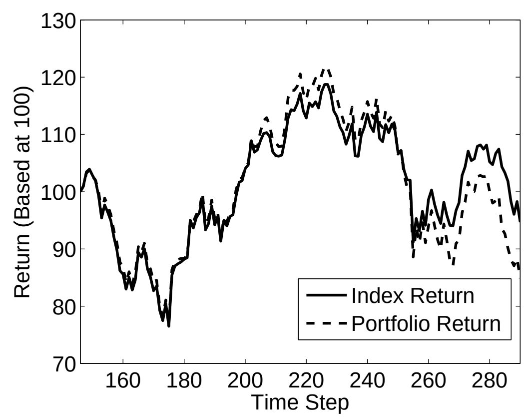
Table 7.5: Ratio of Ex-Post TE over Ex-Ante TE.

Problem Index	Mean	Std Deviation
PORT1	1.2022	0.0546
PORT2	2.0724	0.3437
PORT3	1.0592	0.1374
PORT4	1.8568	0.2779
PORT5	1.7304	0.2742

Nevertheless, this does not exclude ex-ante TE applicability as an objective function for MOEITO. Figure 7.8 compares the return series of a randomly chosen tracker portfolio against its underlying index for PORT2 and PORT5. Despite weak generalization capability in these data sets, the returns deviations were noticeably small at the initial time steps before widening subsequently. In fact, the selected tracker portfolio in PORT5 was able to track almost half of the initial test series before deviating significantly from the underlying index. While this validates the applicability of ex-ante TE as an objective function in these data sets, rebalancing will be required to be more frequent, ultimately translating to higher transaction costs.



(a) PORT2



(b) PORT5

Figure 7.8: Return series of the tracker portfolio and the index in the test data for a randomly chosen algorithmic run.

7.5 Multi-Period Index Tracking

The simulation earlier specifically focused on the generalization capability of MOEITO and the results revealed reasonable tracking performance of the optimized tracker portfolios, though the overall generalization capability depends on the underlying nature of the index, which is beyond the control of the investor. In this section, multi-period index tracking will be considered, representing a more realistic representation of the real-world.

7.5.1 Data Sets & Simulation Setting

The five data sets used in the previous section will be extended here. Their return series (Figure 7.9) generally exhibited an upward trend, except for PORT5. To evaluate MOEITO under different market conditions, PORT6 to PORT10 were created, which are essentially the inverse series of PORT1 to PORT5 respectively. The data sets were divided into training set and test set in the proportion [1, 52] and [53, 291] respectively. A smaller training set was considered to enhance the responsiveness to market changes, while longer test duration allowed the rebalancing strategy to be evaluated fully.

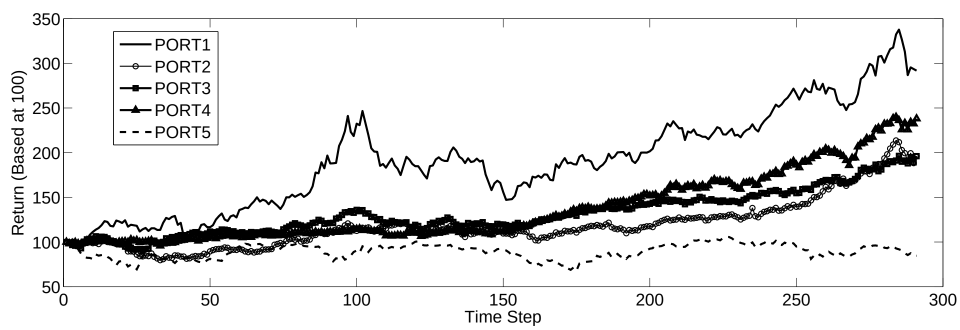


Figure 7.9: Return series for the various data sets based at the initial time step.

Besides measuring ex-post TE, beta based on the test performance was calculated also (7.6). Essentially, it measures the returns correlation of the tracker portfolio and the underlying index, where a beta of 1 will mean that they share similar returns characteristics. Conversely, a tracker portfolio that is more volatile than the underlying index will have a beta with (absolute) value of more than 1 while a tracker portfolio with beta smaller than 1 will mean that it is less sensitive to the index movement. The coefficient of determination, a statistical measure of the goodness of fit of the regression model, will be calculated as a complementary measure for beta.

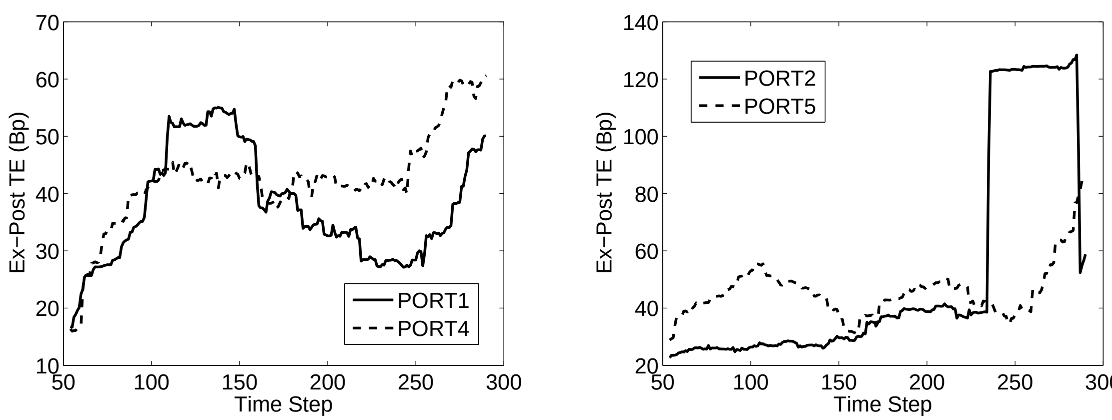
$$\beta = \frac{Cov(r_i, R_i)}{Var(r_i)} \tag{7.6}$$

To assess the operational sustainability of the index tracking system, the net tracker portfolio value in excess to the underlying index at the end of the investment horizon i.e. $F_T - I_T$ will be determined. The expected value should be close to zero if excess returns deviations and unnecessary cash positions are minimized over the entire investment horizon. Annualizing the ratio of the excess portfolio value over the initial invested capital will yield a rough estimate of the management fees required to fund the tracker portfolio, ignoring any compounding effects. While any returns made above the underlying index is entirely secondary, it eliminates the need for management fees and increases the competitiveness of this strategy.

7.5.2 Simulation Result & Discussion

Single-Period Index Tracking

The single-period problem formulation was re-considered here to provide the base case results for the rest of this section. Figure 7.10 plots the ex-post TE attained by the selected tracker portfolio in selected test sets. Although figure 7.10(a) reveals a monotonic rise in ex-post TE straight after the portfolio build-up, this relationship was not generalized to PORT5 in figure 7.10(b), where the ex-post TE remained relatively stable for the entire investment horizon before spiking up in the end. Nevertheless, there are observed spikes in ex-post TE, noticeably for PORT2, due to the absence of active monitoring and rebalancing, which are unacceptable from the perspective of risk management.

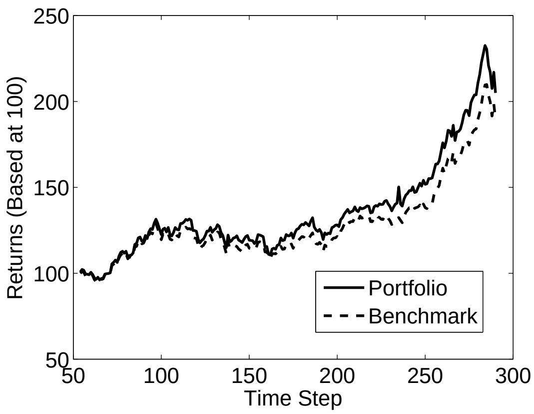


(a) PORT1 and PORT4 (b) PORT2 and PORT5

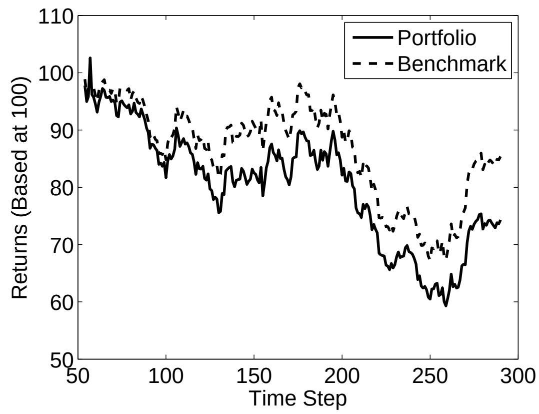
Figure 7.10: Ex-post TE attained without rebalancing.

Examination of the portfolio and index return series (Figure 7.11) revealed that the spikes in ex-post TE can be attributed to sharp changes in index returns, which led to huge returns deviations.

While it might be argued that the deviations are actually positive gains over the index, particularly for the case of PORT2, these gains are clearly not sustainable as negative returns were observed when applied to the corresponding inverse data i.e. PORT7.



(a) PORT2



(b) PORT7

Figure 7.11: Return series of the tracker portfolio and the index in the test data for a randomly chosen algorithmic run.

Rebalancing Strategy

Portfolio rebalancing can be executed periodically or at pre-specified criteria and when triggered, MOEITO will optimize a new tracker portfolio, accounting for the new market conditions and existing holdings. Different rebalancing strategies will be considered here to examine their effects on the overall tracking performance, namely, no rebalancing, periodic rebalancing (i.e. quarterly, semi-annul and annual basis) and event-driven rebalancing when the ex-ante TE exceeds a pre-specified limit.

Table 7.6 summarizes the ex-post tracking performance in data sets, PORT1 to PORT5. Evidently, the average ex-post TE over the investment horizon was generally higher without active rebalancing, together with a lower beta as well. The TE limit in each data set is arbitrarily defined by rounding down the average ex-post TE attained without rebalancing to the nearest tenth basis points. As a result, the proportion of trading days in the investment horizon where the TE limit is exceeded is generally higher without rebalancing. Particularly, this value exceeded 60% in the case of PORT3.

Table 7.6: Tracking Performance in data sets (PORT1-5) for different rebalancing strategies. TE limit for each data set is highlighted in parentheses.

		Without Rebalancing	Annual Rebalancing	Semi- Annual Rebalancing	Quarterly Rebalancing	TE-Limit Rebalancing
PORT1 (40bp)	Total Transaction Costs	226,849	410,527	515,634	645,691	346,166
	Number of Transaction	0.0	4.0	9.0	18.0	2.4
	Average Beta	0.97	0.95	0.94	0.94	0.95
	Regression Coefficient	0.98	0.99	0.99	0.99	0.99
	% of days exceeding TE Limit	34.9%	13.1%	2.7%	0.0%	1.0%
	Average Ex-Post TE	46.18	37.81	34.84	32.65	39.38
PORT2 (70bp)	Total Transaction Costs	215,583	508,409	771,173	1,015,400	1,182,152
	Number of Transaction	0.0	4.0	9.0	18.0	52.0
	Average Beta	1.11	1.03	1.01	1.00	1.08
	Regression Coefficient	0.87	0.88	0.90	0.90	0.87
	% of days exceeding TE Limit	22.3%	22.2%	22.0%	21.9%	21.9%
	Average Ex-Post TE	73.75	70.37	63.05	63.85	71.91
PORT3 (40bp)	Total Transaction Costs	210,415	442,362	630,508	878,675	419,763
	Number of Transaction	0.0	4.0	9.0	18.0	3.4
	Average Beta	1.05	0.96	0.94	0.95	0.95
	Regression Coefficient	0.91	0.95	0.95	0.95	0.94
	% of days exceeding TE Limit	63.5%	13.0%	2.8%	0.1%	1.4%
	Average Ex-Post TE	48.03	39.99	37.88	36.41	39.82
PORT4 (40bp)	Total Transaction Costs	226,138	428,120	571,050	726,027	347,983
	Number of Transaction	0.0	4.0	9.0	18.0	1.9
	Average Beta	1.06	1.01	1.00	1.00	1.03
	Regression Coefficient	0.93	0.97	0.97	0.98	0.96
	% of days exceeding TE Limit	43.0%	2.1%	0.1%	0.0%	0.8%
	Average Ex-Post TE	45.18	35.74	32.24	29.72	38.12
PORT5 (60bp)	Total Transaction Costs	182,778	479,429	758,945	1,117,463	370,782
	Number of Transaction	0.0	4.0	9.0	18.0	2.2
	Average Beta	1.08	0.98	0.97	0.97	1.00
	Regression Coefficient	0.95	0.96	0.97	0.96	0.96
	% of days exceeding TE Limit	23.8%	5.7%	3.0%	1.4%	0.9%
	Average Ex-Post TE	63.13	52.43	48.76	49.98	52.29

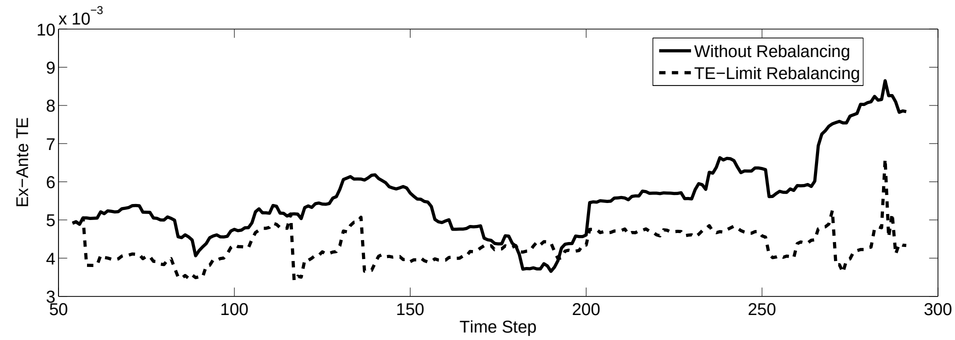
Periodic rebalancing significantly improved tracking performance, where the ex-post TE dropped below the pre-defined TE limit, together with the proportion of investment period exceeding the TE limit. Moreover, a beta closer to 1 was attained together with a higher coefficient of determination. Evidently, the degree of improvement increased with the rebalancing frequency at the expense of higher transaction cost, a direct function of the rebalancing frequency and the number of transactions. Event-triggered rebalancing strategy was able to attune the various data sets. The ex-post TE attained was generally higher with lower transaction costs (except for PORT2) due to infrequent rebalancing as the defined TE limit was probably too high. For completeness, the results for the inverted data sets are shown in Table 7.7. Evidently, similar observations can be drawn from this set of results.

Table 7.7: Tracking Performance in data sets (PORT6-10) for different rebalancing strategies. TE limit for each data set is highlighted in parentheses.

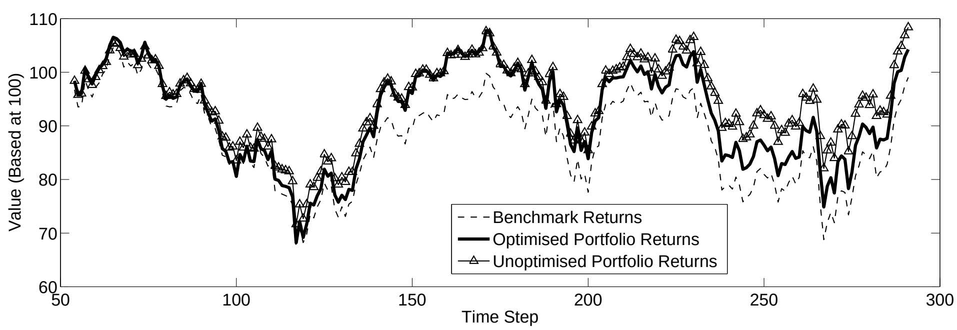
		Without Rebalancing	Annual Rebalancing	Semi- Annual Rebalancing	Quarterly Rebalancing	TE-Limit Rebalancing
PORT6 (40bp)	Total Transaction Costs	226,316	288,433	333,168	397,412	276,229
	Number of Transaction	0	4	9	18	2.4
	Average Beta	1.01	1.01	1.02	1.02	1.02
	Regression Coefficient	0.99	0.99	1	1	0.99
	% of days exceeding TE Limit	48.20%	7.00%	0.30%	0.00%	1.00%
	Average Ex-Post TE	34.25	26.74	25.11	25.09	27.9
PORT7 (70bp)	Total Transaction Costs	213,670	400,058	508,173	631,418	777,988
	Number of Transaction	0	4	9	18	53
	Average Beta	1.03	0.89	0.93	0.93	0.86
	Regression Coefficient	0.82	0.8	0.82	0.82	0.8
	% of days exceeding TE Limit	22.70%	26.20%	22.20%	21.90%	22.30%
	Average Ex-Post TE	72.82	76.43	70.93	69.59	75.77
PORT8 (40bp)	Total Transaction Costs	206,832	348,060	463,039	624,765	344,866
	Number of Transaction	0	4	9	18	3.9
	Average Beta	1	0.96	0.96	0.95	0.97
	Regression Coefficient	0.92	0.94	0.94	0.95	0.94
	% of days exceeding TE Limit	57.00%	13.00%	1.50%	0.10%	1.60%
	Average Ex-Post TE	46.96	41.1	38.38	37.6	40.12
PORT9 (40bp)	Total Transaction Costs	225,533	331,914	410,121	512,188	286,451
	Number of Transaction	0	4	9	18	2
	Average Beta	0.92	0.95	0.96	0.97	0.95
	Regression Coefficient	0.91	0.94	0.95	0.96	0.93
	% of days exceeding TE Limit	55.40%	2.20%	0.40%	0.00%	0.80%
	Average Ex-Post TE	44.93	34.32	30.1	27.15	36.84
PORT10 (60bp)	Total Transaction Costs	190,457	405,164	607,917	899,669	399,060
	Number of Transaction	0	4	9	18	4.2
	Average Beta	1.03	0.91	0.93	0.93	0.93
	Regression Coefficient	0.96	0.97	0.97	0.97	0.97
	% of days exceeding TE Limit	64.00%	14.40%	5.20%	2.10%	1.80%
	Average Ex-Post TE	64.56	51.88	51.82	51.87	50.48

Figure 7.12 compares the time series of the ex-ante TE without rebalancing between TE-limit rebalancing for PORT10 with TE limit of 50bp. The distinct sharp drops in ex-ante TE for the latter whenever it hit 50bp, illustrated the execution of MOEITO. Consequently, the latter was able to attain a lower ex-ante TE throughout the investment horizon, resulting to smaller returns deviations against the underlying index, as shown in Figure 7.12(b) especially for time step 250 onwards. Nevertheless, on a absolute scale, ex-ante TE still remain as a proxy for the expected ex-post TE, where perfect tracking will not be possible, as reflected in Figure 7.12(b) for time step

between 150 and 200.



(a) Ex-Ante TE series



(b) Return series versus underlying index series

Figure 7.12: Comparison of the tracking performance without rebalancing versus TE-limit rebalancing for PORT10 with TE limit of 50bp.

One major challenge of periodic rebalancing is to define a-prior a suitable rebalancing frequency for different underlying indices. Defining a TE limit is analogous to defining the rebalancing frequency, though TE-limit rebalancing will be more adaptive to the underlying index. To investigate this further, TE limit of $\{30, 40, 50, 60, 70\}$ bps were defined for PORT5 and its corresponding inverted series, PORT10, and the tracking performance are summarized in Table 7.8. Clearly, the rebalancing frequency will scale in accordance to the pre-defined TE limit, resulting in better tracking performance while escalating transaction costs. Inevitably, perfect tracking is impossible as the marginal improvement diminished with smaller TE limit i.e. average ex-post TE for PORT5 remained above 40bp even for smaller TE limit.

The operational sustainability of the index tracking system is measured by the net tracker portfolio value in excess to the underlying index at the end of the investment horizon. For the index

Table 7.8: Tracking Performance in PORT5 and PORT10 for different TE limits.

	TE Limit	30	40	50	60	70
PORT5	Total Transaction Costs	1,522,020	690,085	450,960	370,782	332,117
	Number of Transaction	29.2	8.2	3.5	2.2	1.7
	Average Beta	0.95	0.95	0.96	1	1.03
	Regression Coefficient	0.97	0.97	0.97	0.96	0.96
	% of days exceeding TE Limit	12.30%	3.50%	1.50%	0.90%	0.70%
	Average Ex-Post TE	41.63	43.95	47.13	52.29	58.21
PORT10	Total Transaction Costs	1,593,563	646,289	399,060	288,735	251,957
	Number of Transaction	43.5	10.6	4.2	1.9	1.3
	Average Beta	0.94	0.93	0.93	0.92	0.93
	Regression Coefficient	0.97	0.97	0.97	0.96	0.96
	% of days exceeding TE Limit	18.30%	4.40%	1.80%	0.80%	0.50%
	Average Ex-Post TE	46.32	47.72	50.48	53.61	57.19

tracking system to be financially independent, any negative balances should be externally funded by investors in the form of management fees. The implied management fees for the various data sets are summarized in Table 7.9, where a positive management fees implied that the fund is self-sustainable and is able to earn a excess returns over the underlying benchmark. Generally, the excess balances will flip for the inverted series except for PORT 4 and PORT9. Considering the average implied management fees between each data series and its corresponding inverted series, the index tracking system is generally self-sustainable, except in PORT2.

Selection of Tracker Portfolio

As the index tracking optimization problem is formulated here as a multi-objective problem, MOEITO will generate a set of tracker portfolios that are non-dominated against each other. The methodology in determining the appropriate tracker portfolio for the test set is outlined in section III and will be denoted as C-AVE-1. The viability of the proposed selection strategy will be examined in this sub-section by comparing the corresponding tracker performance with other strategies. Specifically, two selection strategies based on the lowest TE and transactions cost will be considered, respectively denoted as C-TE and C-TC. Moreover, a single-objective variant of the MOEITO (denoted as

Table 7.9: Implied Management Fees (Annual).

	Mean	Standard Deviation
PORT1	1.79%	0.73%
PORT2	-19.16%	1.97%
PORT3	0.16%	2.36%
PORT4	-1.35%	2.09%
PORT5	-3.20%	1.70%
PORT6	-0.73%	0.26%
PORT7	2.60%	0.73%
PORT8	-0.14%	0.77%
PORT9	-0.22%	0.72%
PORT10	0.62%	1.36%

SOEITO), focusing solely on TE, will be considered also.

Table 7.10 and 7.11 plot the simulation results for the different selection strategies in the various data sets. A lower TE limit was deliberately chosen so as to enhance the performance differences between the selection strategies. Comparing SOEITO and C-TE, which essentially chose tracker portfolio based on TE, the statistical average of total transaction cost and ex-post TE attained by C-TE dominate that obtained by SOEITO in 6 of the data sets (PORT3, 5, 6, 8, 9, 10) i.e. simultaneously achieving a lower TE and transaction costs, while being dominated only in PORT1. This result highlighted the enhanced search capability of multi-objective problem formulation over single objective, as the former will typically trigger a more diverse exploration of the search space. Interestingly, SOEITO was able to attain a significantly low TE for PORT2 and PORT7, though at the expense of insanely high transaction costs!

Moving on to C-TE and C-TC, their tracking performances clearly epitomize their selection strategy where C-TE generally obtaining a lower TE at the expense of higher transaction cost as compared to C-TC, despite the lower transaction frequency. Comparing the costs per transaction will reflect the differences between these two strategies clearer. The rationale of C-TC having a higher transaction frequency is that the selected tracker portfolio will stick to existing holdings and hence less responsive to changing market conditions. Overall, C-AVE-1 represents a good trade off between C-TC and C-TE, where the tracking performance in half of the data sets (i.e. PORT3, 5, 6, 8, 10) dominate that of C-TE and C-TC.

Table 7.10: Tracking Performance in data sets (PORT1-5) for different selection strategies. TE limit for each data set is highlighted in parentheses.

		SOEITO	C-TE	C-TC	C-AVE-1
PORT1 (30)	Total Transaction Costs	450,000	552,568	458,050	479,112
	Number of Transaction	3.5	4.6	15.7	6.3
	Average Beta	0.95	0.95	0.94	0.95
	Regression Coefficient	0.99	0.99	0.99	0.99
	% of days exceeding TE Limit	1.50%	1.90%	6.60%	2.60%
	Average Ex-Post TE	35.01	35.37	36.72	35.55
PORT2 (60)	Total Transaction Costs	5,300,000	1,284,767	1,352,339	1,225,089
	Number of Transaction	52	52.6	58.4	52.4
	Average Beta	1.06	1.03	1.05	1.07
	Regression Coefficient	0.91	0.87	0.85	0.87
	% of days exceeding TE Limit	21.90%	22.10%	24.50%	22.00%
	Average Ex-Post TE	56.75	70.23	81.1	71.84
PORT3 (30)	Total Transaction Costs	1,013,333	957,979	722,514	622,625
	Number of Transaction	9.1	8.3	35.1	9.5
	Average Beta	0.94	0.95	0.93	0.94
	Regression Coefficient	0.94	0.95	0.95	0.96
	% of days exceeding TE Limit	3.80%	3.50%	14.70%	4.00%
	Average Ex-Post TE	40.15	36.41	36.51	35.29
PORT4 (30)	Total Transaction Costs	500,000	577,241	495,421	443,090
	Number of Transaction	4	4.3	19.8	5
	Average Beta	1	1.01	1.01	1.01
	Regression Coefficient	0.97	0.97	0.97	0.97
	% of days exceeding TE Limit	1.70%	1.80%	8.30%	2.10%
	Average Ex-Post TE	33.27	33.18	35.49	33.55
PORT5 (50)	Total Transaction Costs	640,000	547,651	516,812	450,960
	Number of Transaction	5.4	2.5	9.3	3.5
	Average Beta	0.93	0.95	0.96	0.96
	Regression Coefficient	0.96	0.97	0.97	0.97
	% of days exceeding TE Limit	2.30%	1.10%	3.90%	1.50%
	Average Ex-Post TE	54.44	47.19	49.84	47.13

Table 7.11: Tracking Performance in data sets (PORT6-10) for different selection strategies. TE limit for each data set is highlighted in parentheses.

		SOEITO	C-TE	C-TC	C-AVE-1
PORT6	Total Transaction Costs	380,000	315,126	280,743	276,229
(30)	Number of Transaction	2.8	2.5	8	2.4
	Average Beta	1.01	1.02	1.02	1.02
	Regression Coefficient	0.99	0.99	0.99	0.99
	% of days exceeding TE Limit	1.20%	1.10%	3.40%	1.00%
	Average Ex-Post TE	28.77	28.13	29.7	27.9
PORT7	Total Transaction Costs	5,400,000	803,839	777,582	753,231
(60)	Number of Transaction	53	53	56.2	53
	Average Beta	0.96	0.9	0.84	0.89
	Regression Coefficient	0.88	0.82	0.79	0.81
	% of days exceeding TE Limit	22.30%	22.30%	23.60%	22.30%
	Average Ex-Post TE	58.96	71.07	78.62	73.92
PORT8	Total Transaction Costs	1,236,667	745,773	536,930	500,661
(30)	Number of Transaction	11.4	9.5	37.1	10.5
	Average Beta	0.96	0.96	0.96	0.96
	Regression Coefficient	0.94	0.95	0.95	0.95
	% of days exceeding TE Limit	4.80%	4.00%	15.60%	4.40%
	Average Ex-Post TE	41.74	37.3	36.87	35.69
PORT9	Total Transaction Costs	523,333	421,095	383,651	349,192
(30)	Number of Transaction	4.2	4.2	18.9	5
	Average Beta	0.96	0.96	0.96	0.96
	Regression Coefficient	0.95	0.95	0.94	0.95
	% of days exceeding TE Limit	1.80%	1.80%	7.90%	2.10%
	Average Ex-Post TE	31.06	30.77	33.13	31.43
PORT10	Total Transaction Costs	846,667	582,562	465,461	399,060
(50)	Number of Transaction	7.5	3.5	14	4.2
	Average Beta	0.93	0.92	0.92	0.93
	Regression Coefficient	0.96	0.97	0.96	0.97
	% of days exceeding TE Limit	3.10%	1.50%	5.90%	1.80%
	Average Ex-Post TE	59.6	50.59	53.19	50.48

Stochastic Injection and Withdrawal

To evaluate the tracking capability of the proposed MOEITO in the context of stochastic capital injections, the invested capital will change at a pre-defined a injection probability, τ. Essentially, a random number will be generated at each time step and injection will occur when the generated random is less than the flow probability. Apart from τ, another problem parameter will be defined, α, which controls the amplitude of the injected capital at each instant. Essentially the injected capital at each instant will be a random percentage of the existing capital, drawn from a normal distribution of mean 0 and standard deviation α i.e. $N(0, \alpha)$.τ and α will be arbitrarily set as 0.05 here.

Table 7.12 plots the tracking performance in the various data sets with and without capital injections. A 1% annual management fees was considered. The random injection implicitly increases the rebalancing frequency, increasing the transaction frequency and transaction costs. However, TE was not compromised, in fact, TE actually reduced in most of the data sets i.e. PORT1, 2, 4, 6, 7 and 9. Overall, minimal disruption to the tracking performance was observed.

7.6 Summary

This chapter studied a realistic instantiation of the index tracking optimization problem that accounted for stochastic capital injections and practical real-world transactions cost structures and constraints. The proposed MOEITO, acting as the evolutionary optimization platform of the index tracking system, considered the inherent trade-offs between tracking performance and transaction costs and optimized them simultaneously during the investment horizon. Simulations to validate the proposed approach were conducted and presented, based on five data sets representing equity indices from major financial markets. MOEITO was able to balance tracking performances and transaction costs in out-of-sample test data under the various test scenarios considered. Nevertheless, there remained numerous avenues for future works, which include extending to indices of other asset classes, a prior assessment on the trackability of index, active cash management to generate excess returns, etc.

Table 7.12: Tracking Performance in data sets (PORT1-10) with and without capital injections. TE limit for each data set is highlighted in parentheses.

	PORT1 (30bp)		PORT6 (30bp)	
	w. Injection	w/o. Injection	w. Injection	w/o. Injection
Total Transaction Costs	477,858	677,982	276,332	391,362
Number of Transaction	6.3	15.5	2.4	13.3
Implied Management Fees	4.66%	3.57%	-0.45%	-0.11%
Average Beta	0.95	0.94	1.02	1.02
Regression Coefficient	0.99	0.99	0.99	1
% of days exceeding TE Limit	2.60%	1.30%	1.00%	0.20%
Average Ex-Post TE	35.54	32.62	27.92	25.81
	PORT2 (70bp)		PORT7 (70bp)	
	w. Injection	w/o. Injection	w. Injection	w/o. Injection
Total Transaction Costs	1,233,984	1,498,412	749,349	952,534
Number of Transaction	52.5	61.2	53	61.6
Implied Management Fees	-17.49%	-18.69%	3.05%	3.35%
Average Beta	1.07	1.02	0.89	0.9
Regression Coefficient	0.87	0.88	0.81	0.83
% of days exceeding TE Limit	22.10%	22.00%	22.30%	22.10%
Average Ex-Post TE	71.59	68.86	73.91	68.17
	PORT3 (30bp)		PORT8 (30bp)	
	w. Injection	w/o. Injection	w. Injection	w/o. Injection
Total Transaction Costs	627,864	811,410	499,424	600,119
Number of Transaction	9.6	16.4	10.6	17.5
Implied Management Fees	0.14%	-2.24%	0.66%	1.63%
Average Beta	0.94	0.94	0.96	0.96
Regression Coefficient	0.96	0.96	0.95	0.95
% of days exceeding TE Limit	4.10%	2.20%	4.40%	2.50%
Average Ex-Post TE	35.3	35.33	35.84	36.1
	PORT4 (30bp)		PORT9 (30bp)	
	w. Injection	w/o. Injection	w. Injection	w/o. Injection
Total Transaction Costs	444,022	646,016	349,339	467,545
Number of Transaction	5	13.6	5	13.2
Implied Management Fees	0.90%	1.47%	0.55%	1.13%
Average Beta	1.01	1.01	0.96	0.97
Regression Coefficient	0.97	0.98	0.95	0.96
% of days exceeding TE Limit	2.10%	0.50%	2.10%	0.60%
Average Ex-Post TE	33.55	30.83	31.45	28.11
	PORT5 (60bp)		PORT10 (60bp)	
	w. Injection	w/o. Injection	w. Injections	w/o. Injection
Total Transaction Costs	451,390	921,838	398,312	721,244
Number of Transaction	3.5	13.7	4.2	13.4
Implied Management Fees	-0.96%	-0.82%	0.86%	1.17%
Average Beta	0.96	0.95	0.93	0.92
Regression Coefficient	0.97	0.97	0.97	0.97
% of days exceeding TE Limit	1.50%	0.90%	1.80%	1.00%
Average Ex-Post TE	47.15	47.37	50.38	50.67

Chapter 8

Conclusions

MOEA is a class of stochastic search methods that have been proving to be very efficient and effective in solving sophisticated MOP where conventional optimization tools failed to work well. Its main advantage is its capability to sample multiple candidate solutions simultaneously, a task that most classical MOO techniques are found to be wanting. Since real-world problems typically comprise of several non-commensurable and conflicting objectives, MOEA is finding increasingly application to the diverse fields of engineering, bioinformatics, logistics, economics, finance, and etc, with its development significantly accelerating in the past decade. Nevertheless, existing works related to the application of MOEA for investment portfolio management still pales in comparison to the technical complexities involved in real-world implementations.

8.1 Contributions

This work aims to provide a comprehensive treatment on the design and application of multi-objective evolutionary algorithms to address several key issues involved with investment portfolio management, particularly asset allocation and management styles. These issues have been highlighted and avenues to extend the generic MOEA platform for these purposes were proposed and empirically validated with datasets from actual equity indices in major financial markets.

Chapter 3 proposed an order-based representation for portfolio structure that directly control the type of assets included in the portfolio, hence able to handle more realistic real-world constraints. Empirical studies on its influence on the chromosomal population distribution also revealed its underlying flexibility and extendibility. A set of variations operator specific to the order-based representation was proposed also, instrumental in search space exploration and maintaining the genetic diversity throughout the evolutionary progress. Lastly, a PSO-EA hybrid model with a local search strategy of fine-tuning asset weights of the portfolio and a dynamic archiving strategy with optimization history were proposed. Empirical studies through a modified non-fractional knapsack problem demonstrated their viability to portfolio optimization.

Chapter 4 considered the mean-variance model directly as a multi-objective optimization problem and studied the viability of the proposed MOEA platform. Empirical investigations based on data sets representing equity indices from major financial markets were conducted and reasonable approximations of the corresponding efficient frontiers were attainable. The superiority of the proposed chromosomal representation and the corresponding variation operators over existing works was exhibited also. Lastly, the incorporation of preference-based techniques via the proposed PSO-EA hybrid model demonstrated its capability as a decision support system for portfolio managers in real-world implementation.

Chapter 5 examined the effects of incorporating realistic constraints into the mean-variance model. The two set of constraints considered were namely cardinality constraint with buy-in thresholds, and round-lot constraint with transaction cost. The manner to incorporate these constraints into the problem model and the corresponding constraint handling approaches were presented. The empirical studies aptly demonstrated their effects on the attainable efficient frontiers, adhering closely to logical expectations.

Chapter 6 extended MOEA for TTS optimization with maximal total returns and minimal risk exposure. The building blocks for the trading agents were popular technical indicators used commonly in real-world practices and evolutionary platform allowed the examination of their trading characteristics. The Pareto front obtained accurately depicted the inherent tradeoff between risk and returns and the analysis of the TI composition along the risk-return frontier revealed their underlying characteristics and varying degrees of significance in different regions of the tradeoff surface. Lastly, the correlation study suggested that the returns from the test and training data are not correlated in this context, contrary to popular beliefs. Instead, higher returns in training data only corresponded to larger volatility in the returns generated in the test data.

Chapter 7 studied a realistic instantiation of the index tracking optimization problem that accounted for stochastic capital injections and practical real-world transactions cost structures and constraints. The proposed MOEITO was able to balance the inherent tradeoff between tracking performance and transaction costs throughout the investment horizon. The TE-limit rebalancing strategy was instrumental in improving the operational aspects of the index tracking system. Tracking performance and transaction costs in out-of-sample data, based on five data sets representing equity indices from major financial markets, further affirmed its viability in real-world implementation.

8.2 Future Works

Despite covering several aspects of investment portfolio management under the evolutionary multi-objective optimization platform, these works barely scratched the surface of what is left to be addressed. Many operational issues remain to be sorted out and further extensive validations are absolutely necessary before the proposed models can be extended to real-world implementation. For example, the empirical analysis was based exclusively on the data set provided by Beasley [11]. Clearly, it will be imperative to further evaluate the proposed evolutionary platform under a larger range of data sets in the near future. In addition, computational efficiency was considered only in terms of fitness evaluations. Since the number of evaluations will not be the true case of efficiency in real-world implementation, another immediate consideration will be the actual computational time complexity. This will be especially relevant for portfolios involved in real-time electronic trading and rebalancing.

One major criticism on the mean-variance model is the implicit buy-and-hold investment strategy assumed in this single-period optimization problem, which is in direct conflict to the ever-changing market conditions. It will definitely be instructive to consider multi-period or continuous portfolio optimization model via dynamic correlation models and/or enhance the predicative capability of the model by replacing historical expected returns with forecasted returns instead. The latter can leverage on previous works pertaining to time series forecasting under a MOEA platform [48].

The realistic constraints considered in this work have been delicately isolated so that their individual and combined impact on the attainable efficient frontier can be studied in greater detail. Nevertheless, extending to real-world implementation will definitely warrant the simultaneous consideration of the various constraints and stress testing of the evolutionary platform capability in this aspect is absolutely necessary in future work.

On TTS optimization, the empirical analysis revealed the intrinsic characteristics of the TI considered, corresponding to different risk-return profile. Future work should certainly involve extending similar analysis to other TI that are able to detect other market signal not covered by the current group of TI. Moreover, the empirical studies should be extended to other data sets to further validate the evolutionary model.

In the aspect of index tracking optimization, this thesis has only started the ball rolling with a multi-period instantiation of the problem that considered both tracking performance and transaction costs. The portfolio rebalancing strategy was based on the tracker portfolio deviation with respect to the underlying index. This model can be further enhanced by incorporating TTS to assess which index constituents/existing holdings should be acquired / liquated at each rebalancing instances, thus avoiding unnecessary slippage of asset value in the portfolio. In addition, the independency of the objective functions with respect to the structure of the proposed MOEITO allows perfect substitutability to other objective functions that can offer better predicative capability and operational efficiency, which is definitely a promising avenue for future research.

Bibliography

[1] D. H. Ackley and M. L. Littman, "Interactions between learning and evolution," *Artificial Life II*, pp. 478-507, 1992.

[2] M. Akian, J. L. Menaldi and A. Sulem, "On an investment-consumption model with transaction costs," *SIAM Journal on Control and Optimization archive*, vol. 34, no. 1, pp. 329364, 1996.

[3] S. S. Alexander, "Price Movements in Speculative Markets: Trends or Random Walks," *Industrial Management Review*, vol. 2, no. 2, pp. 7-26, 1961.

[4] F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules," Working Paper, The Wharton School, University of Pennsylvania, 1995.

[5] F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules," *Journal of Financial Economics*, vol. 51, no. 2, pp. 245-271, 1999.

[6] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 84-89, 1998.

[7] R. Armananzas and J. A. Lozano, "A multiobjective approach to the portfolio optimization problem," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1388-1395, 2005.

[8] S. Arnone, A. Loraschi and A. Tettamanzi, "A genetic approach to portfolio selection," *Neural Network World*, vol. 3, no. 6, pp. 597-604, 1993.

[9] P. Di Barba, M. Farina and A. Savini, "An improved technique for enhancing diversity in Pareto evolutionary optimization of electromagnetic devices," *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 20, no. 2, pp. 482-496, 2001.

[10] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York, 1996.

[11] J. E. Beasely, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, pp. 1067-1072, 1990.

[12] J. E. Beasley, N. Meade and T. J. Chang, "An evolutionary heuristic for the index tracking problem," *European Journal of Operational Research*, vol. 148, pp. 621643, 2003.

[13] L. Becker and M. Seshadri, "GP-evolved technical trading rules can outperform buy and hold," *In Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing*, 2003.

[14] R. K. Belew, J. McInerney and N. N. Schraudolph, "Evolving Networks: Using the Genetic Algorithm with Connectionist Learning," *Artificial Life II*, pp. 511-547, 1992.

[15] P. L. Bernstein and A. Damodaran, Investment Management, John Wiley & Sons, 1998.

[16] D. Bienstock, "Computational study of a family of mixed-integer quadratic programming problems," *Mathematical Programming*, vol. 74, no. 2, pp. 121 - 140, 1996.

[17] F. Black, "Capital Market Equilibrium with Restricted Borrowing," *The Journal of Business*, vol. 45, no. 3, pp. 444-455, 1972.

[18] P. Bosman and D. Thierens, "The Balance between Proximity and Diversity in Multi-objective Evolutionary Algorithms," *IEEE Transaction on Evolutionary Computation*, vol. 7, no. 2, pp.174-188, 2003.

[19] P. Brandimarte, Numerical Methods in Finance and Economics: A MATLAB-Based Introduction (Statistics in Practice), Wiley-Interscience; 2nd edition, 2006.

[20] J. Branke and H. Schmeck, "Designing evolutionary algorithms for dynamic optimization problems," in Theory and Application of Evolutionary Computation: Recent Trends, (eds.) S. Tsutsui and A. Ghosh, Springer-Verlag, pp. 239-262, 2002.

[21] J. Branke, "Evolutionary Approaches to Dynamic Optimization Problems Updated Survey," in *Proceedings of the GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pp. 27-30, 2001.

[22] J. Branke. Evolutionary Optimization in Dynamic Environment. Kluver Academic Publishers, University of Karlsruhe, Germany, 2001.

[23] I. Branke, T. Kauler, C. Schmidt, and H. Schmeck, "A multipopulation approach to dynamic optimization problems," in *Adaptive Computing in Design and Manufacturing 2000*, ser. LNCS. Berlin, Germany: Springer-Verlag, 2000.

[24] J. Branke, "Evolutionary approaches to dynamic environments - a survey," *in GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pp. 134-137, 1999.

[25] H. Braun and P. Zagorski, "ENZO-M - A Hybrid Approach for Optimizing Neural Networks by Evolution and Learning," *Parallel Problem Solving from Nature III*, pp. 440-451, 1994.

[26] K. M. Bretthauer and B. Shetty, "The nonlinear knapsack problem - algorithms and applications," *European Journal of Operational Research*, vol. 138, no. 3, pp. 459-472, 2002.

[27] W. Brock, J. Lakonishok and B. LeBaron, "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns," *Journal of Finance*, vol. 47, no. 5, pp. 17311764, 1992.

[28] S. J. Brown, W. N. Goetzmann and A. Kumar, "The Dow Theory: William Peter Hamiltons Track Record Reconsidered," *Journal of Finance*, vol. 53, no. 4, pp. 1311-1333, 1998.

[29] S. Browne, “Beating a moving target: optimal portfolio strategies for outperforming a stochastic benchmark,” *Finance and Stochastics*, vol. 3, pp. 275294, 1999.

[30] X. Cai, N. Zhang, G. K. Venayagamoorthy and D. C. Wunsch, “Time series prediction with recurrent neural networks using a hybrid PSO-EA algorithm,” in *Proceedings of 2004 IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 1647-1652, 2004.

[31] X. Cai and D. C. Wunsch, “Engine data classification with simultaneous recurrent network using a hybrid PSO-EA algorithm,” in *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 2319-2323, 2005.

[32] J. Y. Campbell, “Have Individual Stocks Become More Volatile? An Empirical Exploration of Idiosyncratic Risk,” *Journal of Finance*, vol. 56, no.1, pp. 1-43, 2001.

[33] J. Y. Campbell, A. W. Lo and A. C. MacKinlay, The Econometrics of Financial Markets, Princeton, NJ: Princeton University Press, 1997.

[34] W. Cedeno and V. R. Vemuri, “On the use of niching for dynamic landscapes,” in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 361-366, 1997.

[35] M. C. Chan, C. C. Wong, B. K. S. Cheung and G.Y. N. Tang, “Genetic algorithms in multi-stage asset allocation system,” in *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 2002.

[36] J. P. Chancelier, B. ksendal and A. Sulem, “Combined Stochastic Control and Optimal Stopping, and Application to Numerical Approximation of Combined Stochastic and Impulse Control,” Department of Mathematics, University of Oslo, 2000.

[37] T. J. Chang, N. Meade, J. E. Beasley and Y. M. Sharaiha, “Heuristics for cardinality constrained portfolio optimization,” *Computers and Operations Research*, vol. 27, pp. 1271-1302, 2000.

[38] C. Y. Cheong, K. C. Tan and B. Veeravalli, “Solving the exam timetabling problem via a multi-objective evolutionary algorithm a more general approach,” *IEEE Symposium on Computational Intelligence in Scheduling 2007*, Honolulu, HI, USA, April 1-5, pp. 165-172, 2007.

[39] S. H. Chen and C. H. Yeh, “Toward a computable approach to the efficient market hypothesis: An application of genetic programming,” *Journal of Economic Dynamics and Control*, vol. 21, no. 6, pp. 1043-1063, 1997.

[40] W. Chen, R. Zhang, W. Zhang, “Management of Stochastic Portfolio using Improved Genetic Algorithm,” in *Proceedings of the Sixth World Congress on Intelligent Control and Automation*, vol. 1, pp. 3557-3561, 2006.

[41] W. Chen, R. T. Zhang, Y. M. Cai and F. S. Xu, “Particle Swarm Optimization for Constrained Portfolio Selection Problems, ” in *Proceedings of the 2006 International Conference on Machine Learning and Cybernetics*, pp. 2425-2429, 2006.

[42] W. Chen, R. T. Zhang and W. G. Zhang, “Stochastic portfolio model and its application for genetic algorithms,” in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3486- 3489, 2005.

[43] Q. Chen and S. Guan, "Incremental multiple objective genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 3, pp. 1325-1334, 2004.

[44] S. C. Chiam, K. C. Tan and A. A. Mamun,"Investigating technical trading strategy via an multi-objective evolutionary platform." *Expert Systems with Applications*, vol. 36, no. 7, pp. 10408-10423, 2009.

[45] S. C. Chiam, K. C. Tan, C. K. Goh and A. A. Mamun, "Improving locality in binary representation via redundancy," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, accepted, 2008.

[46] S. C. Chiam , K. C. Tan and A. A. Mamun, "A Memetic Model of Evolutionary PSO for Computational Finance Applications," *Expert Systems With Applications*, accepted, 2008.

[47] S. C. Chiam, K. C. Tan and A. Al Mamum, "Evolutionary multi-objective portfolio optimization in practical context," *International Journal of Automation and Computing*, vol. 5, no. 1,pp. 67-80, 2008.

[48] S. C. Chiam, K. C. Tan and A. Al. Mamun, "Multiobjective Evolutionary Neural Networks for Time Series Forecasting," in *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization*, pp. 346-360, 2007.

[49] S. C. Chiam, A. Al. Mamun and Y. L. Low, "A Realistic Approach to Evolutionary Multiobjective Portfolio Optimization," *Proceedings of the IEEE Congress on Evolutionary Computation*, 2007.

[50] H. G. Cobb, "An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments, " Technical Report AIC-90-001, Naval Research Laboratory, Washington, DC, 1990.

[51] C. A. Coello Coello, D. A. V. Veldhuizen and G. B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems ,Kluwer Academic / Plenum Publishers, 2002.

[52] C.A.C. Coello, "Handling Preferences in Evolutionary Multiobjective Optimization: A Survey," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 30-37, 2000

[53] J. P. Cohoon, S. U. Hegde, W. N. Martin and D. Richards, "Floorplan design using distributed genetic algorithms," in *Proceedings of the IEEE International Conference on Computer Aided-Design*, pp. 452-455, 1988.

[54] R. W. Colby and T. A. Meyers, The encyclopedia of technical market indicators, Dow Jones-Irwin, 1988.

[55] F. Corielli and M. Marcellino, "Factor based index tracking, " *Journal of Banking & Finance*, vol. 30, pp. 22152233, 2006.

[56] P. Cortez, M. Rocha and J. Neves, "A Meta-Genetic Algorithm for Time Series Forecasting," in *Proceedings of the 10th Portuguese Conference on Artificial Intelligence*, pp. 21-31, 2001.

[57] Y. Crama and M. Schyns, "Simulated annealing for complex portfolio selection problems," in *European Journal of Operational Research*, vol. 150, no. 3, pp. 546-571, 2003.

[58] M. A. Dashti, Y. Farjami, A. Vedadi, M. Anisseh, "Implementation of particle swarm optimization in construction of optimal risky portfolios, " in *Proceedings of the I2007 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 812-816, 2007.

[59] M. H. A. Davis and A. R. Norman, "Portfolio selection with transaction costs," *Mathematics of Operations Research*, vol. 15, no. 4, pp. 676713, 1990.

[60] K. A. De Jong, An analysis of the behaviour of a class genetic adaptive systems, Ph.D thesis, University of Michigan, 1975.

[61] K. Deb, J. Sundar, U. Bhaskara and S. Chaudhuri,, "Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273-286, 2006 .

[62] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, New York, 2001.

[63] R. S. Dembo, J. M. Mulvey and S.A. Zenios, "Large-Scale Nonlinear Network Models and Their Application," *Operations Research*, vol. 37, no. 3, pp. 353-372,1989.

[64] V. Devireddy and P. Reed, "Efficient and Reliable Evolutionary Multiobjective Optimization Using ϵ-Dominance Archiving and Adaptive Population Sizing" in *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, pp. 130-131, 2004.

[65] L. Diosan, "A multi-objective evolutionary approach to the portfolio optimization problem," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 2, pp. 183-187, 2005.

[66] C. Dose and S. Cincotti, "Clustering of financial time series with application to index and enhanced index tracking portfolio," in *Physica A: Statistical Mechanics and its Applications*, vol. 355, no. 1, pp. 145-151, 2005.

[67] K. Dowd, "Adjusting for risk: An improved Sharpe ratio," *International Review of Economics & Finance*, vol. 9, no. 3, pp. 209-222, 2000.

[68] G. Dozier, W. Britt, M. P. SanSoucie, P. V. Hull, M. L. Tinker, R. Unger, S. Bancroft, T. Moeller and D. Rooney, "Evolving High-Performance Evolutionary Computations for Space Vehicle Design," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2201-2207, 2006.

[69] B. Dumas and E. Luciano, "An exact solution to a dynamic portfolio choice problem under transaction costs," *Journal of Finance*, vol. 48, no. 2, pp. 577595, 1991.

[70] J. Eastham and K. Hastings, "Optimal impulse control of portfolios," *Mathematics of Operations Research*, vol. 13, pp. 588605, 1988.

[71] M. Ebner, M. Shackleton and R. Shipman, "How Neutral Networks Influence Evolvability," *Complexity*, vol. 7, no.2, pp. 19-33, 2001.

[72] M. Ehrgott, K. Klamroth and C. Schwehm, "An MCDM Approach to Portfolio Optimization," *European Journal of Operational Research*, vol. 155, pp. 752-770, 2004.

[73] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer, 2003.

[74] E. J. Elton and M. J. Gruber, Modern Portfolio Theory and Investment Analysis , John Wiley & Sons; 5th edition, 1995.

[75] R. M. Escudero, R. R. Torrubiano and A. Suarez, "Selection of Optimal Investment Portfolios with Cardinality Constraints," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 8551-8557, 2006.

[76] L. J. Eshelman and J. D. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incest," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 115-122, 1991.

[77] A. A. A. Esmin, G. Lambert-Torres and G. B. Alvarenga, "Hybrid Evolutionary Algorithm Based on PSO and GA Mutation," in *Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*, pp. 57, 2006.

[78] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *Journal of Finance*, vol. 23, pp. 383 -417, 1970.

[79] E. F. Fama and M. E. Blume, "Filter Rules and Stock-Market Trading," *Journal of Business*, vol. 39, no. 1, pp. 226-241, 1966.

[80] Y. Fang and S. Y. Wang, "A fuzzy index tracking portfolio selection model," *International conference on Computational Science*, vol. 3516, 2005.

[81] M. M. Farsangi, H. Nezamabadi-Pour and K. Y. Lee, "Multi-objective VAr Planning with SVC for a Large Power System Using PSO and GA," in *Proceedings of the IEEE PES Power Systems Conference and Exposition*, pp. 274-279, 2006.

[82] J. E. Fieldsend, J. Matatko and M. Peng, "Cardinality constrained portfolio optimization," in *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning*, pp. 788-793, 2004.

[83] J. C. Ferreira, C. M. Fonseca and A. Gaspar-Cunha, "Methodology to select solutions from the pareto-optimal set: a comparative study," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation 2007*, pp. 789-796, 2007.

[84] L. J. Fogel, A. J. Owens, M. J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley, 1966.

[85] R. French and A. Messinger, "Genes, phenes and the Baldwin Effect: Learning and evolution in a simulated population," *Artificial Life IV*, pp. 277-282, 1994.

[86] A. Frino and D. Gallagher, "Tracking S&P 500 index funds," *The Journal of Portfolio Management*, vol. 28, no. 1, pp. 4455, 2001.

[87] S. M Focardi and F. J Fabozzi, "A methodology for index tracking based on time-series clustering," *Quantitative Finance*, vol. 4, no. 4, pp. 417-425, 2004.

[88] C. Fyfe, J. P. Marney and H. F. E. Tarbert, "Technical analysis versus market efficiency - a genetic programming approach," *Applied Financial Economics*, vol. 9, no. 2, pp. 183-191, 1999.

[89] A. A Gaivoronski, S. Krylov and N. V. D. Wijst, "Optimal portfolio selection and dynamic benchmark tracking," *European Journal of Operational Research*, vol. 163, no. 1, pp. 115-131, 2005.

[90] A. A Gaivoronski and G. Pflug, "Value-at-Risk in Portfolio Optimization: Properties and Computational Approach," *Technical Report*, Norwegian University of Science and Technology, 2000.

[91] A. L. Garcia-Almanza and E. P. K. Tsang, "Simplifying Decision Trees Learned by Genetic Algorithms," *in Proceedings of the Congress on Evolutionary Computation*, pp. 7906-7912, 2006.

[92] M. Gerrits and P. Hogeweg, "Redundant code of an NP-complete problem allows effective Genetic Algorithm search," in *Proceedings of the 1st Parallel Problem Solving from Nature*, vol. 496, pp. 70-74, 1991.

[93] M. Gilli and E. Kellezi, "Threshold Accepting for Index Tracking," University of Geneva, 2001.

[94] J. J. Grefenstette, "Genetic algorithms for changing environments, "in *Proceedings of the Second International Conference on Parallel Problem Solving from Nature*, pp. 137-144, 1992.

[95] F. Grimaccia, M. Mussetta, P. Pirinoli and R. E. Zich, "Optimization of a reflectarray antenna via hybrid evolutionary algorithms," in *Proceedings of the 17th International Zurich Symposium on Electromagnetic Compatibility*, pp. 254 - 257, 2006.

[96] F. Grimaccia, M. Mussetta, P. Pirinoli and R. E. Zich, "Genetical Swarm Optimization (GSO): a class of Population-based Algorithms for Antenna Design," in *Genetical Swarm Optimization (GSO): a class of Population-based Algorithms for Antenna Design*, pp. 467-471, 2006.

[97] E. A. Grimaldi, F. Grimaccia, M. Mussetta, P. Pirinoli and R. E. Zich, "Genetical Swarm Optimization: a New Hybrid Evolutionary Algorithm for Electromagnetic Applications," in *Proceedings of the 18th International Conference on Applied Electromagnetics and Communications*, pp. 1-4, 2005.

[98] C. Grosan, A. Abraham and M. Nicoara, "Performance tuning of evolutionary algorithms using particle sub swarms," in *Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2005.

[99] D. E. Goldberg and R. E. Smith, "Nonstationary function optimization using genetic algorithms with dominance and diploidy," in *Genetic Algorithms*, J. J. Grefenstette, Ed: Lawrence Erlbaum, 1987, pp. 59-68.

[100] D. E. Goldberg, and J. Richardson, "Genetic algorithms with sharing for multi-modal function optimization," in *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41-49, 1987.

[101] F. G. Guimares, F. Campelo, H. Igarashi, D. A. Lowther and J. A. Ramrez, "Optimization of Cost Functions Using Evolutionary Algorithms With Local Learning and Local Search," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1641-1644, 2007.

[102] J. Guo, Y. Wu and W. Liu, "An Ant Colony Optimization Algorithm with Evolutionary Operator for Traveling Salesman Problem," in *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, vol. 1, pp. 385-389, 2006.

[103] B. S. Hadad and C. F. Eick, "Supporting polyploidy in genetic algorithms using dominance vectors," in *Evolutionary Programming*, ser. LNCS, P. J. Angeline et al., Eds. Berlin, Germany: Springer-Verlag, vol. 1213, pp. 223-234, 1997.

[104] W. E. Hart, "Adaptive Global Optimization with Local Search," Ph.D. dissertation, University of California, San Diego, 1994.

[105] K. Hellwig, Bewertung von Ressourcen, Physica Verlag, Heidelberg, 1987.

[106] G. E. Hinton and S. J. Nowlan, "How Learning Can Guide Evolution," *Complex Systems*, vol. 1, pp. 495-502, 1987.

[107] T. Hiroyasu, S. Nakayama and M. Miki,"Comparison Study of SPEA2+, SPEA2, and NSGA-II in Diesel Engine Emissions and Fuel Economy Problem," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 236-242, 2005.

[108] J. H. Holland, Adaptation in Natural Artificial Systems: An Introductory Analysis with Applocations to Biology, Control, and Artificial Intelligence, MIT press, 1992.

[109] J. Horn, "Multicriterion Decision Making," *Multicriterion Decision Making*, vol. 1, pp. F1.9:1-F1.9:15,1997

[110] S. Hwang and S. Satchell, "Tracking Error: ex ante versus ex post measures," *Journal of Asset Management*, vol. 2, no. 3, 2001.

[111] H. Iba and N. Nikolaev, "Financial Data Prediction by Means of Genetic Programming," *in Procedings. of the Sixth International Conference on Computing in Economics and Finance*, 2000.

[112] K. Ikeda, H. Kita and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean nearto optimal?," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 2, pp. 957 - 962, 2001

[113] H. Ishibuchi, T. Yoshida and T.Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204-223, 2003.

[114] N. Jegadeesh, "Evidence of Predictable Behavior of Security Returns," *Journal of Finance*, vol. 45, no. 3, pp. 881898, 1990

[115] N Jegadeesh and S. Titman, "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency," *Journal of Finance*, vol. 48, no. 1, pp. 6591, 1993.

[116] M. C. Jensen, "Problems in Selection of Security Portfolios: The Performance of Mutual Funds in the Period 1945-64," *Journal of Finance*, vol. 6, 1968.

[117] R. Jeurissen and J. van den Berg, "Optimized index tracking using a hybrid genetic algorithm," *IEEE Congress on Evolutionary Computation*, pp. 2327-2334, 2008.

[118] R. Jeurissen, "A Hybrid Genetic Algorithm to track the Dutch AEX-index," *Bachelor Thesis*, Informatics & Economics, Faculty of Economics, Erasmus University Rotterdam, 2005.

[119] R. Jiang and K. Y. Szeto, "Extraction of investment strategies based on moving averages: A genetic algorithm approach," *in Proceedings of the 2003 IEEE International Conference on Computational Intelligence for Financial Engineering*, pp. 403-410, 2003.

[120] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303-317, 2005.

[121] C. F. Juang, "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 2, pp. 997-1006, 2004.

[122] C. F. Juang and Y. C. Liou, "TSK-type recurrent fuzzy network design by the hybrid of genetic algorithm and particle swarm optimization," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2314-2318, 2004.

[123] C. F. Juang and Y. C. Liou, "On the hybrid of genetic algorithm and particle swarm optimization for evolving recurrent neural network," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 3, pp. 2285-2289, 2004a.

[124] B. A. Julstrom, "Redundant genetic encodes may not be harmful," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 791, 1999.

[125] R. Keesing and D. G. Stork, "Evolution and learning in neural networks: the number and distribution of learning trials affect the rate of evolution," *Advances in neural information processing systems 3*, pp. 804-810, 1990.

[126] H. Kellerer, U. Pferschy and D. Pisinger, Knapsack Problems, Springer, 2004.

[127] G. Kendall and Y. Su, "A multi-agent based simulated stock market - testing on different types of stocks," *in Proceedings of the 2003 Congress on Evolutionary Computation*, vol. 4, pp. 2298-2305, 2003.

[128] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[129] J. Kennedy, R. C. Eberhart and Y. Shi, Swarm Intelligence, Morgan Kaufmann, 2001.

[130] B. C. Kho, "Time-varying risk premia, volatility, and technical trading rule profits: Evidence from foreign currency futures markets," *Journal of Financial Economics*, vol. 41, no. 2, pp. 249-290, 1996.

[131] E. F. Khor, K. C. Tan, T. H. Lee, and C. K. Goh, "A study on distribution preservation mechanism in evolutionary multi-objective optimization," *Artificial Intelligence Review*, vol. 23, no. 1, pp. 31-56, 2005.

[132] H. Kitano, "Empirical studies on the speed of convergence of neural network training using genetic algorithms," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 397-404, 1990.

[133] J. D. Knowles, D. W. Corne and M. Fleischer, "Bounded archiving using the Lebesgue measure," in *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, vol. 4, pp. 2490-249, 2003.

[134] J. D. Knowles, and D. W. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100-116, 2003.

[135] H. Konno and H.Yamazaki, "Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market," *Management Science*, vol. 37, no. 5, pp. 519-531 , 1991.

[136] J. Korczak and P. Roger, "Stock timing using genetic algorithms," *Applied stochastic models in business and industry*, vol. 18, no. 2, pp. 121 134, 2002.

[137] J. Korczak and P. Lipinski, "Evolutionary Building of Stock Trading Experts in a Real-Time System," *in Proceedings of the 2004 Congress on Evolutionary Computation*, pp. 940 947, 2004.

[138] P. G. Korning, "Training neural networks by means of genetic algorithms working on very long chromosomes," *International Journal of Neural Systems*, vol. 6, no. 3, pp. 299-316, 1995.

[139] M. Koshino, H. Murata, H. Kimura, "Improved particle swarm optimization and application to portfolio selection, " *Electronics and communications in Japan*, vol. 90, no. 3 pp. 13-25, 2007.

[140] N. Krasnogor, "Studies on the Theory and Design Space of Memetic Algorithms", Ph.D. Thesis, Faculty of Computing, Mathematics and Engineering, University of the West of England, Bristol, U.K, 2002.

[141] K. W. C. Ku, M. W. Mak and W. C. Siu, "Approaches to combining local and evolutionary search for training neural networks: a review and some new results," *Advances in evolutionary computing: theory and applications*, pp. 615-641, 2003.

[142] J.W. Kwiatkowski, "Algorithms for index tracking," *IMA Journal of Management Mathematics*, vol. 4, no. 3, pp. 279-299, 1992.

[143] S. Lajili-Jarjir and Y. Rakotondratsimba, "The number of securities giving the maximum return in the presence of transaction costs," *Quality and Quantity*, vol. 42, no. 5, 2007.

[144] M. W. S. Land, "Evolutionary algorithms with local search for combinatorial optimization," Ph.D. dissertation, Univ. of California, San Diego, 1998.

[145] M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Combining convergence and diversity in evolutionary multi-objective optimization," *Evolutionary Computation*, vol. 10 no. 3, 2002.

[146] B. N. Lehmann, "Fads, Martingales, and Market Efficiency," *Quarterly Journal of Economics*, vol. 105, no. 1, pp. 1-28, 1990.

[147] J. Lewis, E. Hart, and G. Ritchie, "A comparison of dominance mechanisms and simple mutation on nonstationary problems," in *Parallel Problem Solving from Nature*, ser. LNCS, A. E. Eiben, T. Bck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, vol. 1498, pp. 139-148, 1998.

[148] J. Li and E. P. K. Tsang, "Improving Technical Analysis Predictions: An Application of Genetic Programming," *in Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, pp. 108112, 1999.

[149] J. Li, H. Z. Liu, B. Yang, J. B. Yu, N. Xu and C. H. Li, "Application of an EACS algorithm to obstacle detour routing in VLSI physical design," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1553-1558, 2003.

[150] M. Lim and R. J. Coggins, "Optimal trade execution: an evolutionary approach," *in Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1045-1052, 2005.

[151] L. Lin, L. Cao, J. Wang and C. Zhang, "The Applications of Genetic Algorithms in Stock Market Data Mining Optimisation," *in Proceedings of Fifth International Conference on Data Mining, Text Mining and their Business Applications*, pp. 273-280, 2004.

[152] D. Lin, S. Wang and H. Yan, "A multiobjective genetic algorithm for portfolio selection," in *Proceedings of the 5th International Conference on Optimization: Techniques and Applications*, Hong Kong, 2001.

[153] A. W. Lo and A. C. MacKinlay, "When Are Contrarian Profits Due to Stock Market Overreaction?," *Review of Financial Studies*, vol. 3, no. 2, pp. 172205, 1990.

[154] A. Loraschi, A. Tettamanzi, M. Tomassini and P. Verda, "Distributed genetic algorithms with an application to portfolio selection problems," *Artificial Neural Networks and Genetic Algorithms*, pp. 384-387, 1995.

[155] D. G. Luenberger, Investment Science, Oxford University Press, Oxford, 1998.

[156] Y. H. Lui and D. Mole, "The use of fundamental and technical analyses by foreign exchange dealers: Hong Kong evidence," *Journal of International Money and Finance*, vol. 17, no. 3, pp. 535-545, 1998.

[157] L. P. Lukac, B. W. Brorsen and S. H. Irwin, "A Test of Futures Market Disequilibrium Using Twelve Different Technical Trading Systems," *Applied Economics*, vol. 20, no. 5, pp. 623-39, 1988.

[158] S. Mahfoud and G. Mani, "Financial Forecasting using Genetic Algorithms," *Applied Artificial Intelligence*, vol. 10, no. 6, pp. 543-566, 1996.

[159] S.W. Mahfoud, *Niching Methods for Genetic Algorithms*, PhD thesis, University of Illinois at Urbana-Champaign. 1995.

[160] B. Malkiel, "Returns from investing in equity mutal funds 1971 to 1991," *The Journal of Finance*, vol. 50, no. 2, pp. 549572, 1995.

[161] M. Mansini and M. Speranza, "Heuristic algorithms for the portfolio selection problem with minimum transaction lots," *European Journal of Operational Research*, vol. 114, pp. 219233, 1999.

[162] D. Maringer and O. Oyewumi, "Index tracking with constrained portfolios," *International Journal of Intelligent Systems in Accounting and Finance Management*, vol. 15, no. 1-2 , pp. 57 - 71 , 2007

[163] D. Maringer, Portfolio Management with Heuristic Optimization, Advances in Computational Management Science, vol. 8, XIV, 2005.

[164] D. Maringer and H. Kellerer, "Optimization of cardinality constrained portfolios with a hybrid local search algorithm," *OR Spectrum*, vol. 25, no. 4 , pp. 481-495, 2004.

[165] H. Markowitz, Portfolio selection: Efficient diversification of investments, New York, John Wiley & Sons, 1959.

[166] H. Markowitz, "Portfolio Selection," *Journal of Finance*, vol. 7 no. 1 pp. 77-91, 1952.

[167] H. Mausser and D. Rosen, "Applying Scenario Optimization to Portfolio Credit Risk," *Journal of Risk Finance*, vol. 2, no. 2, pp. 36-48, 2001.

[168] N. Meade and G. Salkin, "Developing and Maintaining an Equity Index Fund," *The Journal of the Operational Research Society*, vol. 41, no. 7, pp. 599-607, 1990.

[169] R. C. Merton, "Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case," *The Review of Economics and Statistics*, vol. 51, no. 3, pp. 247-257, 1969.

[170] R. C. Merton, "Optimum consumption and portfolio rules in a continuous-time model," *Journal of Economic Theory*, vol. 3, no. 4, pp. 373-413, 1971.

[171] J. Miao, "Volatility filter for index tracking and long-short market-neutral strategies," *Journal of Asset Management*, vol. 8, pp. 101-111, 2007.

[172] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, 1999

[173] D. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 762-767, 1989.

[174] R. Moral-Escudero, R. Ruiz-Torrubiano and A. Suarez, "Selection of Optimal Investment Portfolios with Cardinality Constraints," *IEEE Congress on Evolutionary Computation*, pp. 8551-8557, 2006.

[175] R. Morrison, Designing Evolutionary Algorithms for Dynamic Environments, Springer-Verlag, 2004.

[176] H. Muhlenbein, M. Schomisch and J. Born, "The Parallel Genetic Algorithm as Function Optimizer," in *Proceedings of the Fourth international Conference on Genetic Algorithms*, pp. 271-278, 1991.

[177] J. Murphy, Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications, Prentice-Hall, 1999.

[178] C. Neely, "Risk-adjusted, ex ante, optimal technical trading rules in equity markets," *International Review of Economics and Finance*, vol. 12, no. 1, pp. 69-87, 2003.

[179] C. Neely, P. Weller and R. Dittmar, "Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach," *Journal of Financial and Quantitative Analysis*, vol. 32, no. 4, pp. 405426, 1997.

[180] J. O, J. Lee, J. W. Lee and B. T. Zhang, "Dynamic Asset Allocation for Stock Trading Optimized by Evolutionary Computation," *IEICE Transactions on Information and Systems*, vol. 88, pp. 1217-1223, 2005.

[181] K. J. Oh, T. Y. Kim and S. Min, "Using genetic algorithm to support portfolio optimization for index fund management, " *Expert Systems with Applications*, vol. 28, pp. 371379, 2005.

[182] N. Okay and U. Akman, "Index tracking with constraint aggregation," *Applied Economics Letters*, vol. 10, no. 14, pp. 913-916, 2003.

[183] B. ksendal and A. Sulem, "Optimal Consumption and Portfolio with Both Fixed and Proportional Transaction Costs," Department of Mathematics, University of Oslo, 1999.

[184] Y. S. Ong, M. H. Lim, N. Zhu and K. W. Wong, "Classification of Adaptive Memetic Algorithms: A Comparative Study," *IEEE Transactions On Systems, Man and Cybernetics - Part B*, vol. 36, no. 1, pp. 141-152, 2006.

[185] Y. S. Ong and A. J. Keane, "Meta-Lamarckian in Memetic Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99-110, 2004.

[186] K. W. C. Ku, M. W. Mak and W. C. Siu, "Approaches to combining local and evolutionary search for training neural networks: a review and some new results," *Advances in evolutionary computing: theory and applications*, pp. 615-641, 2003.

[187] Y. S. Ong, Artificial Intelligence Technologies in Complex Engineering Design, Ph.D. Thesis, School of Engineering Science, University of Southampton, United Kingdom, 2002.

[188] Y. Orito, H. Yamamoto and G. Yamazaki, "Index fund selections with genetic algorithms and heuristic classifications," *Computers & Industrial Engineering*, vol. 45, pp. 97109, 2003.

[189] D. Orvosh and L. David, "Shall we repair? Genetic algorithms, combinatorial optimization, and feasibility constraints," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 650, 1993.

[190] A. Osyczka and S. Krenich, "Evolutionary Algorithms for Multicriteria Optimization with Selecting a Representative Subset of Pareto Optimal Solutions," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 141-153, 2001.

[191] J. R. Perez and J. Basterrechea, "Comparison of Different Heuristic Optimization Methods for Near-Field Antenna Measurements," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 549-555, 2007.

[192] G. R. Raidl and J. Gottlieb, "Empirical Analysis of Locality, Heritability and Heuristic Bias in Evolutionary Algorithms: A Case Study for the Multidimensional Knapsack Problem," *Evolutionary Computation*, vol. 13, no. 4, pp. 441-475, 2005.

[193] Y. Rahmat-Samii, "enetic algorithm (GA) and particle swarm optimization (PSO) in engineering electromagnetics," in *Proceedings of the ICECom 17th International Conference on Applied Electromagnetics and Communications*, pp. 1-5, 2003.

[194] N. Raman and F. B. Talbot, "The job shop tardiness problem: A decomposition approach," *European Journal of Operational Research*, vol. 69, no. 2, pp. 187-199, 1993.

[195] M. J Ready, "Profits from Technical Trading Rules," *Financial Management*, vol. 31, no. 3, pp. 43-61, 2002.

[196] I. Rechenberg, Evolutionsstrategie, Frommann-Holzboog, 1994.

[197] F. K. Reilly and K. C. Brown, Investment Analysis and Portfolio Management, South-Western College Publication, 2002.

[198] M. Riepe and M. Werner, "Are Enhanced Index Mutual Funds Worthy of Their Name?" *Journal of Investing*, vol. 7, no. 2, pp. 6-15, 1998.

[199] J. Robinson, S. Sinton and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna," *IEEE Antennas and Propagation Society International Symposium*, vol. 1, pp. 314-317, 2002.

[200] R. T. Rockafellar and S. Uryasev, "Optimization of Conditional Value-at-Risk," *Journal of Risk*, vol. 2, no. 3, pp. 21-41, 2000.

[201] S. Ronald, J. Asenstorfer and M. Vincent, "Representational redundancy in evolutionary Algorithms," in *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation*, vol. 2, pp. 631-636, 1995.

[202] F. Rothlauf and D. E. Goldberg, "Redundant Representations in Evolutionary Computation," *Evolutionary Computation*, vol. 11, no. 4, pp. 381-415, 2003.

[203] J. Rowe, K. Vinsen and N. Marvin, "Parallel GAs for Multiobjective Functions," in *Second Nordic Workshop on Genetic Algorithms and Their Applications*, pp. 61-70, 1996.

[204] G. Rudolph and A. Agapie, "Convergence Properties of Some Multi-Objective Evolutionary Algorithms,"in *Proceedings of the 2000 Conference on Evolutionary Computation*, pp. 1010-1016, 2000.

[205] G. Rudolph, "On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set," in *Proceedings of the 1998 Conference on Evolutionary Computation*, pp. 511-516, 1998.

[206] R. Ruiz-Torrubiano and A. Surez, "Use of Heuristic Rules in Evolutionary Methods for the Selection of Optimal Investment Portfolios," *IEEE Congress on Evolutionary Computation*, pp. 212-219, 2007.

[207] C. Ryan, "Diploidy without dominance," in *Proceedings of the 3rd Nordic Workshop Genetic Algorithms*, pp. 63-70, 1997.

[208] H. Sato, H. E. Aguirre and K. Tanaka, "Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs," in *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization*, Matsushima, Japan, March 5-8, pp. 5-20, 2007.

[209] J. D. Schaffer, "Multi-Objective Optimization with Vector Evaluated Genetic Algorithms," in *Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100, 1985.

[210] C. Schoreels, B. Logan and J.M. Garibaldi, "Agent based genetic algorithm employing financial technical analysis for making trading decisions using historical equity market data," *in Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 421-424, 2004.

[211] C. Schoreels and J. M. Garibaldi, "A comparison of adaptive and static agents in equity market trading," *in Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 393-399, 2005.

[212] B. Scherer and R. Douglas, Introduction to Modern Portfolio Optimization with NUOPT and S-Plus, Springer, 2005.

[213] M. Schroder, "Optimal Portfolio Selection with Fixed Transaction Costs: Numerical Solutions," Working Paper,Michigan State University, 1995.

[214] K. Shahookar and P. Mazumder, "A genetic approach to standard cell placement using metagenetic parameter optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 5, pp, 500-511, 1990.

[215] J. Shapcott, "Index Tracking: Genetic Algorithms for Investment Portfolio Selection," *Edinburgh Parallel Computing Centre*, EPCCSS9224, 1992.

[216] W. F. Sharpe, "The Sharpe Ratio," *Journal of Portfolio Management*, vol. 21,no. 1, pp. 4958, 1994.

[217] W. F. Sharpe, "Mutual Fund Performance," *Journal of Business*, vol. 39, no. 1, pp. 119-138, 1966.

[218] J. Shoaf and J. A. Foster, "The efficient set GA for stock portfolios," in *Proceedings of the 1998 IEEE International Conference on Computational Intelligence*, pp. 354 - 359, 1998.

[219] S. Shreve and H. M. Soner, "Optimal investment and consumption with transaction costs," *The Annals of Applied Probability*, vol. 4, pp. 609692, 1994.

[220] A. J. Skinner and J. Q. Broughton, "Neural networks in computational materials science: training algorithms," *Modelling and Simulation in Material Science and Engineering*, vol. 3, pp. 371-390, 1995.

[221] P. Skolpadungket, K. Dahal and N. Harnpornchai, "Portfolio Optimization Using Multi-Objective Genetic Algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 516-523, 2007.

[222] P. Skolpadungket and K. Dahal, "A Survey on Portfolio Optimisation with Metaheuristics," in *Proceedings of the International conference on software ,Knowledge,Information Management and Applications*, pp. 176-184, 2006

[223] E. Sorenson, K. Miller and V. Samak, "Allocating between active and passive management," *Financial Analysts Journal*, vol. 54, no. 5, pp. 1831, 1998.

[224] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1994.

[225] F. Streichert, H. Ulmer and A. Zell, "Evolutionary algorithms and the cardinality constrained portfolio selection problem," *Operations Research Proceedings*, 2003.

[226] F. Streichert and M. Tanaka-Yamawaki,"The Effect of Local Search on the Constrained Portfolio Selection Problem," *IEEE Congress on Evolutionary Computation*, pp. 8537-8543, 2006.

[227] F. Streichert, H. Ulmer and A. Zell, "Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem," *Genetic and Evolutionary Computation*, vol. 3103, pp. 1239-1250, 2004.

[228] F. Streichert, H. Ulmer and A. Zell, "Evaluating a Hybrid Encoding and Three Crossover Operators on the Constrained Portfolio Selection Problem," *Congress on Evolutionary Computation*, vol. 1, pp. 932-939, 2004.

[229] R. Subbu, P. P. Bonissone, N. Eklund, S. Bollapragada and K. Chalermkraivuth, "Multiobjective Financial Portfolio Design: A Hybrid Evolutionary Approach," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1722-1729, 2005.

[230] Y. Tabata and E. Takeda, "Bicriteria Optimization Problem of Designing an Index Fund," *The Journal of the Operational Research Society*, vol. 46, no. 8, pp. 1023- 1032, 1995

[231] H. Takehara, "An interior point algorithm for large scale portfolio optimization," *Annals of Operations Research*, vol.45, no. 1-4, pp. 373-386, 1993.

[232] K. C. Tan, C. K. Goh, Y. J. Yang, and T. H. Lee, "Evolving better population distribution and exploration in evolutionary multi-objective optimization," *European Journal of Operational Research*, vol. 171, no. 2, pp. 463-495, 2006.

[233] T. Z. Tan, C. Quek and G. S. Ng, "Brain-inspired genetic complementary learning for stock market prediction," *in the Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2653-2660, 2005.

[234] K. C. Tan, Y. H. Chew and L. H. Lee, "A hybrid multi-objective evolutionary algorithm for solving vehicle routing problem with time windows," *Computational Optimization and Applications*, vol. 34, pp. 115-151, 2006.

[235] K. C. Tan, C. Y. Cheong and C. K. Goh, "Solving multi-objective vehicle routing problem with stochastic demand via evolutionary computation," *European Journal of Operational Research*, vol. 177, pp. 813-839, 2007.

[236] M. P. Taylor and H. Allen, "The use of technical analysis in the foreign exchange market," *Journal of International Money and Finance*, vol. 11, no. 3, pp. 304314, 1992.

[237] J. Tobin , The theory of Portfolio Selection, in The Theory of Interest Rates, ed. by F. Hahn and F. Brechling. Macmillan & Co. Ltd, London, 1965

[238] J. Tobin, "Liquidity Preference as Behavior Towards Risk," *The Review of Economic Studies*, vol. 25, no. 2, pp. 65-86, 1958.

[239] A. Torn and A. Zilinskas, "Global optimization," *Lecture Notes in Computer Science*, vol. 350, pp. 255, 1989.

[240] R. R. Torrubiano and A. Suarez, "Use of Heuristic Rules in Evolutionary Methods for the Selection of Optimal Investment Portfolios," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 212-219, 2007.

[241] R. K. Ursem, "Mutinational GA optimization techniques in dynamic environments, " in *Proceedings of the 2000 Genetic and Evolutionary Computation Congress*, pp. 19-26, 2000.

[242] F. Vavak, K. Jukes, and T. C. Fogarty, "Adaptive combustion balancing in multiple burner boiler using a genetic algorithm with variable range of local search, "in *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 719-726, 1997.

[243] D. A. V. Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: analysing the state-of-the-arts," *Evolutionary Computation*, vol. 8, no. 2, pp.125-147, 2000.

[244] J. Vesterstrom and R. Thomsen, "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems," in *Proceedings of the 2004 Congress on Evolutionary Computation*, vol. 2, pp. 1980-1987, 2004.

[245] J. H. Wang and S. M. Chen, "Evolutionary stock trading decision support system using sliding window," *in Proceedings of the 1998 IEEE International Conference on Computational Intelligence*, pp. 253-258, 1998.

[246] R. L. Weissman, Mechanical Trading Systems: Pairing Trader Psychology with Technical Analysis, Wiley Trading, 2005.

[247] F. M. Werner, D. Bondt and R. H. Thaler, "Further Evidence on Investor Overreaction and Stock Market Seasonality," *Journal of Finance*, vol. 42, no. 3, pp. 557-581, 1987.

[248] D. L. Whitley, V. S. Gordon and K. E. Mathias, "Lamarckian Evolution, The Baldwin Effect and Function Optimization," *Parallel Problem Solving from Nature III*, pp. 6-15, 1994.

[249] M. Wineberg and F. Oppacher, "Enhancing the GAs ability to cope with dynamic environments," in *Proceedings of the 2000 Genetic and Evolutionary Computation Congress*, pp. 3-10, 2000.

[250] P. Wolfe, "The Simplex Method for Quadratic Programming," *Econometrica*, vol. 27, no. 3, pp. 382-398, 1959.

[251] Y. Xia, S. Wang and X. Deng, "A compromise solution to mutual funds portfolio selection with transaction cost," *A compromise solution to mutual funds portfolio selection with transaction cost*, vol. 134, no. 3, pp. 564-581, 2001.

[252] Y. Xia, B. Liu, S. Wang and K. K. Lai, "A model for portfolio selection with order of expected returns," *Computers & Operations Research*, vol. 27 no. 5, pp. 409-422, 2000.

[253] F. Xu, W. Chen and L. Yang, "Improved Particle Swarm Optimization for Realistic Portfolio Selection, " in *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 185-190, 2007.

[254] S. Yang and R. Tinos, "Hyper-selection in dynamic environments," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pp. 3184-3191, 2008.

[255] X. Yao and Y. Liu, "A New Evolutionary System for Evolving Artificial Neural Networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 694-713, 1997.

[256] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp.1423-1447, 1999.

[257] L. Zhang, R. Raut and Y. Jiang, "A novel evolutionary algorithm for analog VLSI layout placement design," in *Proceedings of the 2nd Annual IEEE Northeast Workshop on Circuits and Systems*, pp. 117-120, 2004.

[258] H. Zheng, A. Wong and S. Nahavandi, "Hybrid ant colony algorithm for texture classification," in *Proceedings of the Congress on Evolutionary Computation*, vol. 4, pp. 2648-2652, 2003.

[259] Z. Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane and K. Y. Lum, "Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization," *IEEE Transactions On Systems, Man and Cybernetics - Part C*, vol. 37, no. 1, pp. 66-76, 2007.

[260] Z. Z. Zhou, Y. S. Ong, M. H. Lim and B. S. Lim, "Memetic Algorithm using Multi-Surrogates for Computationally Expensive Optimization Problems," *Soft Computing Journal*, vol. 11, no. 10, pp. 957-971, 2007.

[261] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.

[262] E. Zitzler and L. Thiele, "Multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

[263] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. Fonseca, "Performance assessment of multi-objective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117-132, 2003.

[264] A. Zorin and A. Borisov, "Traditional and Index Tracking Methods for Portfolio Construction by Means of Neural Networks," *Scientific Proceedings of Riga Technical University*, Information Technology and Management Science, 2002.