INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS MONTERREY

SCHOOL OF ENGINEERING
DIVISION OF MECHATRONICS AND INFORMATION TECHNOLOGIES
GRADUATE PROGRAMS

TECNOLÓGICO DE MONTERREY®

DOCTOR OF PHILOSOPHY
in
INFORMATION TECHNOLOGIES AND COMMUNICATIONS
MAJOR IN INTELLIGENT SYSTEMS

An Evolutionary Risk Optimization Algorithm to Implement Investment Strategies in a Multi-Period Framework with Dynamic Restrictions

By

Rubén Anton Aguilar Rivera

DECEMBER 2015

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Division of Mechatronics and Information Technologies
Graduate Program

The committee members, hereby, certify that have read the dissertation presented by Rubén Anton Aguilar Rivera and that it is fully adequate in scope and quality as a partial requirement for the degree of **Doctor of Philosophy in Information Technologies and Communications**, with a major in **Intelligent Systems**.

Dr. Manuel Valenzuela Rendón
ITESM Campus Monterrey
Principal Advisor

Dr. Carlos A. Coello Coello
CINVESTAV-IPN
Committee Member

Dr. Hugo Terashima Marín
ITESM Campus Monterrey
Committee Member

Dr. José de Jesús Rodríguez Ortiz
ITESM Campus Monterrey
Committee Member

Dr. Osmar H. Zavaleta Vázquez
EGADE Business School
Committee Member

Dr. Jorge Welti Chanes
Director of Research and Graduate Programs
School of Engineering

Copyright Declaration

I, hereby, declare that I wrote this dissertation entirely by myself and, that, it exclusively describes my own research.

Rubén Anton Aguilar Rivera
Monterrey, N.L., México
December 2015

Dedication

I dedicate this work to the memory of Antonio, Guadalupe, Jesús, Santiaga, and the rest of them.

Acknowledgements

I thank Tecnológico de Monterrey, Research Group with Strategic Focus on Intelligent Systems of the National School of Engineering and Sciences, for the support on tuition. Besides, I thank the Consejo Nacional de Ciencia y Tecnología (CONACyT) for the support for living through the PNPC program. Also, I thank to my advisor, Dr. Valenzuela, who helped me clarify my ideas about this work. I thank Dr. Rodriguez and Dr. Gordillo, who help me enrolling to this academic program. I thank Dr. Coello, Dr. Terashima, and Dr. Zavaleta for being part of the committee and their comments about my work. I thank my parents who supported me in different ways. Finally, I thank you, the reader, for your time. I hope you find this work useful.

An Evolutionary Risk Optimization Algorithm to Implement Investment Strategies in a Multi-Period Framework with Dynamic Restrictions

by

Rubén Anton Aguilar Rivera

Abstract

Financial investment is an important economic activity. The value of indexes like the Dow Jones Industrial Average (DJI), the Standard & Poor 500 (S&P500) or domestic stock market indexes are commonly used as a measure of a country's level of development. Financial markets provide a comfortable method to generate profit from diverse industries and commercial activities. Nevertheless, investors should consider also uncertainty in stock prices, legal restrictions, and transaction costs when making decisions.

Even when several works have been published about ways to deal with the difficulties described above, many investors continue relying only on their own experience to make decisions. The limitations of the current approaches and their complexity have caused investors to overlook their benefits. Therefore, they are in need of tools to help them make correct decisions in practical situations.

Investors are continuously concerned with making the best possible decision. From the wide range of available methods, portfolios have the advantage of including the uncertainty of the decisions (i.e. risk) into the optimization process. Besides, they provide a set of optimal solutions and an explanation about how investors choose a portfolio according with their preferences. Utility functions are used to model this behavior. Nevertheless, the inclusion of new restrictions to the problem definition prevents the application of traditional solution methods. Moreover, the risk metric is restricted to the covariance matrix of the asset's returns only. Finance theory has identified these drawbacks and proposed solutions based on a multi-period definition of the problem, where a time horizon is considered instead of a static definition of the market.

Nevertheless, this work has identified the following limitations to multi-period portfolio optimization approaches: They are limited to optimization of the portfolio's return from the last period of time only; they rely on theoretical utility functions to describe the investor's preference; finally, the overlook the information provided by data innovations arriving during the time horizon. This work assumes this information is useful to make better investment decisions.

The review indicated the multi-period definition of the problem is developed using dynamic programming, which allow the inclusion of transaction costs and other state-dependent restrictions to it. Nevertheless, its solution has proved to be a difficult task. Multi-period theory references are mainly concerned with finding closed-form solutions to the problem for a given combinations of dynamic restrictions, risk metrics and utility functions. Definition of sub-problems is a common solution technique. On the other hand, evolutionary algorithms

have been mainly applied to solve static portfolio optimization problems. Round-lots and compulsory assets are some examples. The conclusion was the application of evolutionary algorithms to solve multi-period portfolio optimization problems has received limited attention in the literature.

This work introduces an investment method based on multi-period portfolio theory implemented with evolutionary algorithms. A Monte-Carlo approach is proposed to handle dynamic restrictions without the complications of purely mathematical methods. Transactions costs, portfolio unbalance, and inflation are the ones considered. Moreover, an identification process of the particular investor's preference is presented to avoid the use of theoretical utility models. Also, the method considers data innovations to evaluate the current state of the market to allow adaptive decisions. The solutions model is divided in two parts: A multi-objective stochastic optimization evolutionary algorithm to solve multi-period portfolio problems, and the Investment Strategies method which uses the information about the market state, investor's preference, and portfolio performance to make decisions.

The method has the advantage to include dynamic restrictions, which are usually not included in the optimization process of traditional methods. The most important restriction are transaction costs, because the profit obtained by trading can be severely decimated by them. Also, the method includes a procedure to identify the investor's particular preference, therefore, it makes decisions closer to the investor's expectations. The method is fully automatic, providing regular investors with a useful tool to find investment recommendations. Although, the method is to be further enhanced with the inclusion of static restrictions and trading execution capabilities to have a complete investing system.

The proposed method was tested with real data from American and Mexican markets and was compared against buy-and-holds and single-period optimal portfolios, which are common methods used by investors. The experiments considered the following performance metrics: Maximum loss, total time to reach the investor's goal, final portfolio's return, number stop loss occurrences, expected return and risk, and the Sharpe's ratio. Statistic analysis concluded the proposed method outperformed the others for the proposed metrics. The Investment Strategies method showed to have lower maximum losses and higher Sharpe's ratios than the other methods. Besides, the results indicate Investment Strategies dominate other methods when expected return and risk are considered. A significant difference was found between the results form the American market and the ones from the Mexican market. Finally, differences were found in the results obtained with different risk metrics.

The results concluded the American market was subject of higher risk than the Mexican market. The analysis of the results concluded good investment decisions come from a balance between transaction and following the trends of the market. Also, different information sources should be considered when making decisions. The method is subject to improvements. For example, other methods could be used instead of normal multi-variate distributions to simulate the returns. Also, dynamic investment strategies could be devised to adapt the behavior of the algorithm to the current market scenario.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Investment is the sacrifice of current wealth for the sake of profit in the future (Sharpe, Alexander, & Bailey, 1999). Risk is inherent to investment because of uncertainty about the final outcome of decisions. Every endeavor is an investment of valuable resources like time, money, or manpower. It is a game with actual consequences where people put their well-being at stake.

Man has always been concerned with making the best possible decision. For example, operations research was developed to apply advanced mathematical techniques to decision making in a wide range of environments. The present work is focused on financial investment, which has place at financial markets. Markets are a relatively safe environment where an investor can increase his wealth without the technical knowledge of a particular trade.

Nevertheless, financial investing presents its own difficulties. Markets are non-stationary environments where dynamic effects and costs cannot be ignored. On the contrary, traditional finance theory is limited to a static model of the investment phenomenon. The literature (presented in later chapters) shows some of the efforts to include dynamic restrictions into the problem. Although most of these solutions can only be applied under restrictive conditions. The need of practical methods was identified in this research.

Evolutionary algorithms (EAs) have been successfully applied to solve complex problems in different fields. They have the advantage of handling the problem restrictions indirectly, including them into the objective function. Most of the available approaches are concerned with finding closed-form solutions to investment problems with dynamic restrictions. Nevertheless, these difficulties can be avoided using evolutionary algorithms.

This chapter briefly presents the motivation, the problem statement, and the solution model to solve the investment problem with dynamic restrictions. These points are further developed later in this work. The main contributions are enlisted at the end of the chapter.

1.1 Motivation

Financial investment is an important economic activity. The value of indexes like the Dow Jones Industrial Average (DJI), the Standard & Poor 500 (S&P500) or domestic stock market indexes are commonly used as a measure of a country's level of development. Financial markets provide a comfortable method to generate profit from diverse industries and commercial activities. Nevertheless, investors should consider also uncertainty in stock prices, legal restrictions, and transaction costs when making decisions.

Even when several works have been published about ways to deal with the difficulties described above, many investors continue relying only on their own experience to make decisions. The limitations of the current approaches and their complexity have caused investors to overlook their benefits. Therefore, they are in need of tools to help them make correct decisions in practical situations.

Dynamic restrictions are crucial because time is needed to attain any significant level of profit. A simple solution would be choosing a data frequency according to the investor's preferences. For example, a highly risk-averse investor could prefer make daily transactions even when the expected return is low. On the other hand, annually data deals with higher return values, but at a higher risk. Nevertheless, investors rarely limit themselves to a single decision. Instead, they keep a set of promising securities (i.e. portfolio) for weeks or months and make adjustments when they deem it necessary.

Dynamic approaches model this behavior and find an optimal solution given an utility function. Although, these methods showed to be too complex for everyday use. This have kept them at the "ivory tower" and away from practical applications (Brandt & Santa-Clara, 2006).

A common limitation is the inability to compute the investor's utility function, which is used to describe his preferences about risk and return. References usually assume a theoretical utility model and explode their properties to provide a closed-form solution. They are not concerned with a method to compute the investor's particular preference. Nevertheless, it seems possible to propose a method where an explicit utility function is not required for the task. The necessary information could be provided in parametrically form instead.

Another limitation of dynamic approaches is they do not usually consider innovations, which are data that become available within the time horizon of the investment. These methods do not include innovations because they are concerned with the probability distribution of return of the last period only. The fact a single realization shows poor performance does not mean the method is incorrect. Although, real investors are concerned with current performance. The inclusion of innovations can provide information which can be useful to modify the decisions accordingly to their results. The present work concluded an evolutionary computing method will allow solving investment problems including dynamic restrictions, parametrically investor's preference, and data innovations.

1.2 Problem Statement and Context

There are different types of investments (Sharpe et al., 1999). For example, real investment is when wealth is used to buy tangible assets to make profit. Examples of real investment are the buying of land, tools, or factories. Nevertheless, the success in a particular kind of industry requires time, knowledge, and specific ability. On the other hand, indirect investment occurs at financial markets. They provide a way to invest in a trade without the need of specific knowledge about it. The most common financial instruments are securities. These type of contracts give their holders ownership over a company, besides some rights and dividends. The profit usually comes from trading them at stock markets. The newly issued securities are sold at primary markets where auction mechanisms are used to trade shares. These shares can be bought and sold later at secondary markets.

There are other financial instruments besides securities. The following are some examples: Treasury bills and bonds are examples of fixed-income assets. Their face plate prices are payed to their holders at maturity time. Although, they can also be traded at secondary markets at variable prices. Commodities, like gold or silver, are used to diversify investment, allowing a better control of risk. Futures are contracts to buy or sell a specified instrument or commodity at a given date. Derivatives are contracts which value depends on the performance of some other financial instrument. All of them are subject of trade at their respective markets.

Financial markets are important centers of economic activity. For example, the New York stock exchange (NYSE) had an estimated monthly value of $\$1.55$ USD trillions and a year-to-date value of $\$14.72$ USD trillions at October, 2015 (*WFE Monthly Reports*, 2015, December). The large amount of money traded at financial markets has caused the misconception a fortunate investor could attain extraordinary profit in no time and without effort. This has proved to be far from the truth. The investor should face the uncertainties of financial environments, besides legal restrictions and transaction costs. An investor who blindly ignores these hindrances could end up in bankruptcy. These ideas are summarized in the efficient market model proposed by (Fama, 1970b). An efficient market is the one where all the information is available to every investor at the same time. Historical price data is all the information considered in the weakest form of this model. The efficient market hypothesis states abnormal profit opportunities are impossible because every investor will make the same (optimal) decision. Although, The limited reasoning principle (Russel & Norvig, 2010) states available information could be too vast to be fully processed. Therefore, investors could make (globally) incorrect decisions even at efficient markets. The conclusion is methods to handle information effectively could lead to make profitable investment decisions.

1.2.1 Basic Terminology

Profit is usually presented in the form of rates of return, which allows expressing it independently from the asset price. Besides, the probability distribution of compound returns is normal even when the asset price probability distribution is not (Meucci, 2010). The return of a particular security $r(t)$ is defined as

$$r(t) = \frac{p(t) - p(t-1)}{p(t-1)}, \tag{1.1}$$

where $p(t)$ is security current price at time t. Continuously compound return is defined as

$$r(t) = \ln\left(\frac{p(t)}{p(t-1)}\right). \tag{1.2}$$

According with the model explained by (Sharpe et al., 1999), future security prices are related as

$$E(p(t+1)|\Phi(t)) = [1 + E\left(r(t+1)|\Phi(t)\right)]\, p(t), \tag{1.3}$$

where $\Phi(t)$ is the information set at time t. Equation 1.3 computes the expected value of the future price $p(t+1)$ given the available information. This value depends of the current price value $p(t)$ and the expected value of the future return given the information. In later chapters, equation 1.3 is implemented with a Monte-Carlo method to estimate the outcome of investment decisions based on simulated returns data with a known distribution.

1.2.2 Approaches to Investment Decision Problems

The methods to help the investor make better investment decisions can be split in three groups: Fundamental analysis methods, technical analysis methods, and methods based on modern portfolio theory (Sharpe et al., 1999). They are explained below.

Fundamental Analysis

Fundamental analysis is concerned with security valuation. There are two approaches to it. The first approach attempts to determine the true price of securities. This can be done using the company's internal information and the estimation of certain economic factors. Econometric models like the capital allocation price model (CAPM) are useful for this end (Sharpe et al., 1999). The objective is to determine if a security is mispriced and use this information to make profitable decisions (i.e. arbitrage). For example, a raise in the price of securities is expected when the analysis indicate the asset is under priced at the market. The second approach makes an estimation of the security's return for a given period of time. This information can be used to determine which securities are reliable for a given time period. Probabilistic forecasting models are part of this category.

The drawback of this approach is the difficulty to obtain the necessary information to apply it. For example, the data can be incomplete or not properly formatted. The same occurs to the data about the econometric factors. For example, CAPM model indicates the existence of the market portfolio, which is composed by all the securities of the market. The task to determine it is impossible without suitable simplifications.

Technical Analysis

Technical analysis is concerned with using price history to make investment decisions. Also, volume information is often used by these methods. Technical analysis relies on the existence of patterns in data which repeat frequently. The identification of such patterns allows the prediction of trends which are useful to make profitable decisions. Technical analysis is composed by a wide range of approaches. Chart analysis, Bollinger bands, and moving averages are some examples. Some reviews about them can be found in the literature (Lo, Mamaysky, & Wang, 2000).

Modern Portfolio Theory (MPT)

H. Markowitz (1952) was the first to integrate risk and utility to investment problems. A portfolio **w** is defined in this context as

$$\mathbf{w}(t) = \begin{bmatrix} w_i(t) \\ \vdots \\ w_m(t), \end{bmatrix} \tag{1.4}$$

restricted to

$$\sum_{i=1}^{m} w_i(t) = 1. \tag{1.5}$$

Modern portfolio theory characterizes securities by their expected return $E\left[r_i(t)\right]$ and risk $\sigma_i(t)$. The latter is obtained computing the standard deviation of the returns. Moreover, the expected return and risk of a portfolio is obtained form their components using

$$E\left[r_w(t)\right] = \sum_{i=1}^{m} w_i(t) E\left[r_i(t)\right], \tag{1.6}$$

$$\sigma_w^2(t) = \mathbf{w}^T(t) \begin{bmatrix} \sigma_{11} & \dots & \sigma_{1m} \\ \vdots & \ddots & \vdots \\ \sigma_{m1} & \dots & \sigma_{mm} \end{bmatrix} \mathbf{w}(t). \tag{1.7}$$

The matrix shown in equation 1.7 is the portfolio covariance matrix. Equations 1.6 and 1.7 have an averaging effect over the expected value of returns and the risk of the securities. This is called *portfolio diversification*.

H. M. Markowitz, Lacey, Plymen, Dempster, and Tompkins (1994) reported the critical line method (CLM) to find the optimum portfolio for securities with known $E\left[r_w(t)\right]$ and $\sigma_w(t)$. This method finds the highest return portfolios for a given level of risk. The set of optimal portfolios (for different risk levels) is called the *efficient frontier*, which is the Pareto front formed when both risk and return are simultaneously optimized. Figure 1.2 shows the efficient frontier from securities of the Dow Jones industrial average (2005–2012) and compares it to the risk and return of the individual securities. The highest risk security is always part of the Pareto front.

Utility is used to explain how an investor chooses a portfolio from the efficient frontier. The analysis starts from the assumption that the investor is risk-averse and always chooses the lowest risk portfolio for a given level of return. Utility is related to the "satisfaction" that decisions provide to the investor. Figure 1.1 shows an example of a utility function. Higher levels of return provide higher utility levels, but its rate decreases accordingly. There will be a point where the increase will no longer be attractive to the investor.

Utility functions can be mapped to the risk-return plane in the form of *indifference curves*. These curves represent the combinations of risk-return which have the same utility for the investor. It is possible to show that only one indifference curve will intersect the efficient frontier at a single point (Sharpe et al., 1999). This point represents the optimal portfolio that satisfies the investor's expectations. An example is shown in figure 1.2.

1.2.3 Multi-Period Portfolio Selection Problem (MPPSP)

Traditional portfolio theory is used by financial professionals in combination with fundamental and technical analysis to give investment advice. Nevertheless, it has important limitations. For example, it assumes fractions of securities can be traded, but in practice only round-lots are allowed. Besides, it does not takes into account the impact of transactions on prices, which can be affected by changes of the supply and demand of securities. Also, traditional portfolio theory assumes the probability distributions of stocks are time invariant. In practice, this is

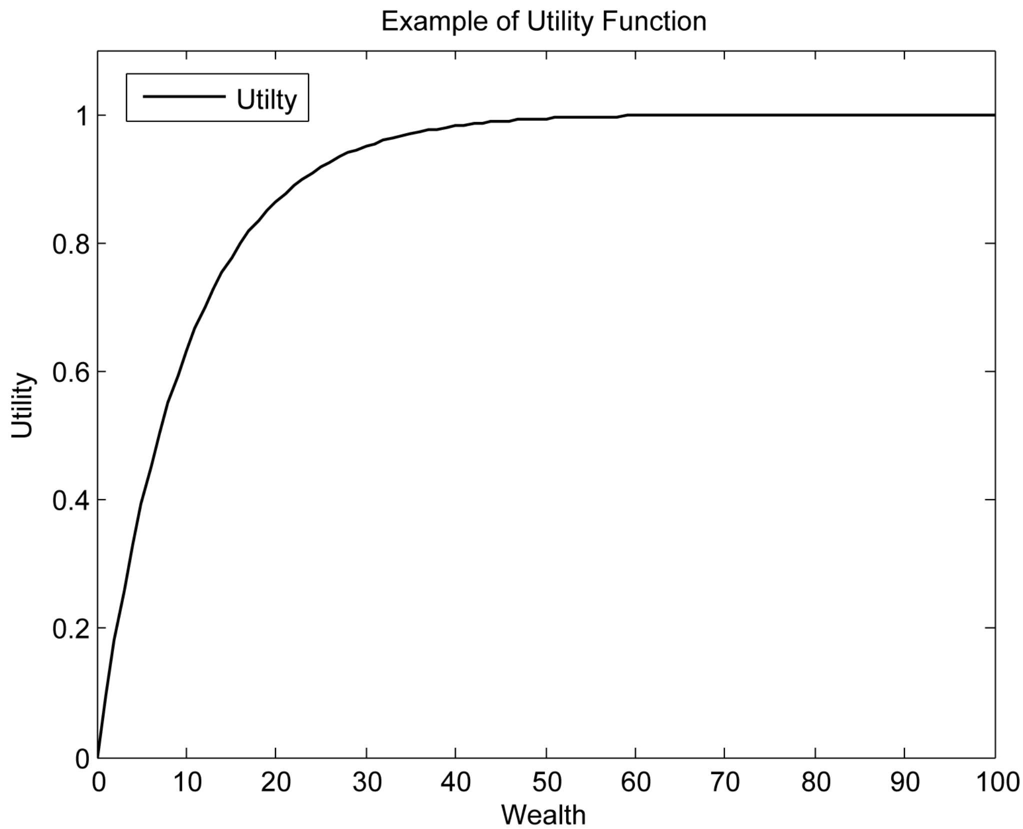

Figure 1.1: An utility function of a risk-averse investor.

seldom the case (Hamilton, 1994). Some real-world limitations (e.g. transaction costs) are overlooked by it. The traditional approach allows the investor choosing a set of optimal portfolios, but the final wealth will differ from their calculations because transaction costs were not considered. Financial professionals used to recommend decisions with high return before transaction costs. There is no guarantee these solutions are optimal when transaction costs are included in the model. Inflation is also a factor to be considered when making practical investment decisions.

In early references, some debate was held about the existence of the multi-period portfolio selection problem. Fama (1970a) proposed a multi-period model where a time horizon T was considered into the optimization process. The decisions seek to optimize the expected value of return at time T instead of present time. A solution was found using dynamic programming. The conclusion was that there is no difference among risk-averse single-period strategies and multi-period strategies under the assumptions of no transaction costs and a perfect market. A. H. Chen, Jen, and Zionts (1971) proposed an extension of the Markowitz's model to multiple periods which allows portfolio revision. The problem was defined with dynamic programming techniques. This approach considered multiple-period scenarios and transaction costs. Nevertheless, its solution was devolved to future works.

Elton and Gruber (1974a) stated single-period and multi-period problems are different

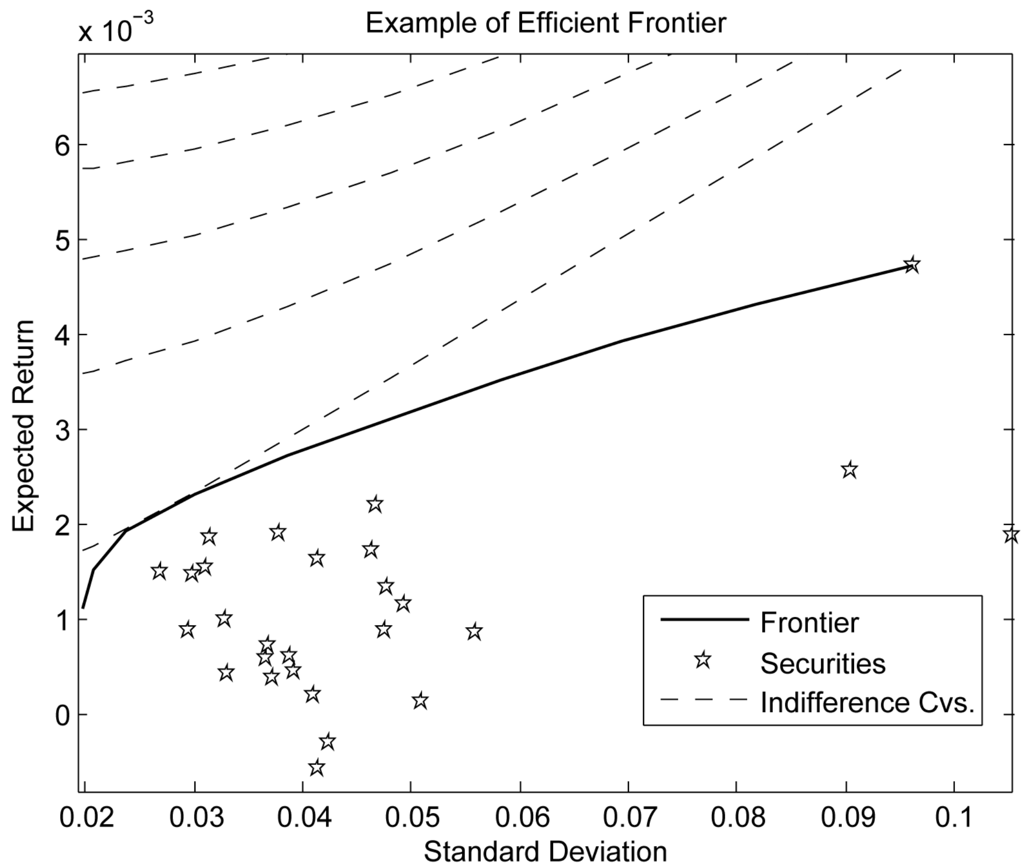

Figure 1.2: Efficient frontier, individual securities, and indifference curves.

when the investor utility function is state-dependent. For example, the inclusion of inflation in the model will cause the same amount of money to have different utility levels at different time periods. At inflationary economies, utility will depend on the portfolio, time, and the interest rate of inflation. The difference will hold even when transactions costs are not considered. Unbalance changes the portfolio's composition along with changes in the market. Transaction costs are directly affected by it. The present work is focused on these dynamic restrictions.

In conclusion, single-period portfolio selection problems becomes multi-period when state-dependency and dynamic restrictions are considered in the model. Examples of these restrictions are inflation and transaction costs. The solution of a multi-period portfolio selection problems is the optimal set of T future decisions given the current information. The complete solution is found at initial time. Data innovations are not relevant to the current multi-period approaches. The solutions found in the literature change with the restrictions and the utility functions considered. The state-of-the-art section will describe some of the approaches to find closed-form solutions to multi-period problems. Also, it presents solutions based on evolutionary algorithms.

1.2.4 Problem Statement

The problem is to find a method with the ability of making investment decisions with dynamic restrictions like transaction costs and portfolio unbalance. Also, it should be able to include the investor's particular preference, avoiding the use of theoretical utility functions. The method should consider data innovations, and use that information to modify recommendations according with the result of the decisions. The desired method should be based on evolutionary algorithms.

1.2.5 Objective of this work

The objective of this work is investigate about what information is useful to make good investment decisions. Dynamic restrictions provide information about the market's nature. Parametric identification of the investor's preference intends to provide further information about his preferences about risk and return. Data innovations provide information about the non-stationary behavior of the market. This work proposes the development of an investment method based on portfolios which includes the information from these sources. The method should have the ability of handling dynamic restrictions and include the investor's preference without using theoretical utility functions. Besides, it should include the information of data innovations to allow decision adaptation based on the current state of the portfolio. Experiments will be conducted to determine if the information from these sources is truly useful to make good investment decisions.

1.3 Research Questions

The problem statement implies evolutionary algorithms have advantageous properties to overcome the limitations explained above. These properties are summarized in the hypothesis statement below. The research questions summarize the problems addressed to accomplish the task. Besides, they remark the importance of the method to test the algorithm and obtain solid conclusions.

1.3.1 Hypothesis Statement

A Multi-period approach to investment problems can provide better recommendations than a single-period approach because the former may handle dynamic restrictions like transaction costs, unbalance, and other state-dependent factors. Moreover, the investor's preferences can be included into the optimization process without relying on theoretical utility functions. Also, information about the current state of the market changes and the method's performance is relevant and useful to make investment decisions.

1.3.2 Research Questions List

- How can the investor's preference be included in the optimization process?
- Can utility functions be excluded from the portfolio selection problem?

- Which parameters could capture the information represented by utility functions?
- Is there inherent differences between the nature of different markets? How do the affect the optimization process?
- What differences among markets are instrumental when performing multi-period portfolio optimization?
- How can an evolutionary algorithm be implemented to solve multi-period portfolio selection problems?
- How should be measured the performance of financial strategies?
- How the proposed method could be appealing to regular investors?

1.4 Solution Overview

Traditional portfolio theory states investors strive to find portfolios with the highest return possible for a given level of risk. Portfolios should be optimal for these competing objectives. Multi-objective optimization has been addressed using evolutionary algorithms in the literature: The second version of the non-domination sorting genetic algorithm (NSGA-II) is a good example (Deb, Pratap, Agarwal, & Meyarivan, 2002). Evolutionary algorithms usually keep a population of candidate solutions which are cleverly combined to find the optimum. Besides, multi-objective evolutionary algorithms use populations to search different solutions simultaneously. These properties make them a suitable approach to solve multi-period portfolio selection problems.

References indicate multi-period portfolio selection problems are difficult to solve. Dynamic programming solution models require the revision of a large number of possible scenarios. In this particular case, valid options do not decrease with time because the possibility of changing from current portfolio to any other is always present. In the literature, solution methods rely on the properties of a particular utility functions or on the definition of subproblems to restrict the valid scenarios. This is the reason why these solutions are valid under certain conditions only.

Some authors have indicated numerical methods are useful to overcome these limitations. Desai, Lele, and Viens (2003) proposed a Monte-Carlo algorithm to find the solution for one of these theoretical models. A Monte-Carlo approach can be easily integrated into an evolutionary algorithm. Although, this approach makes the objective function stochastic. The use of evolutionary algorithms could probably overcome other problems like the parametric inclusion of the investor's preference into the optimization.

Therefore, a new evolutionary algorithm is proposed in this work to solve multi-objective problems with stochastic functions. This problem is more complex than noisy optimization because the level of uncertainty is an objective of the problem that should be accurately estimated. On the contrary, noisy optimization usually conceives noise as a nuisance to be eliminated. The problem treated in this work is closer to robust optimization because the solutions with lower uncertainty are preferred by the investor. A further explanation of these approaches is presented in later chapters.

In the literature, multi-period solutions are only concerned with the optimization the last period return. This work also explored multi-period optimization where sets of portfolios of different time duration are considered. The existence of non-dominated portfolios with heterogeneous time horizons was found and applied to the solve the problem. This approach provides investors with a wider range of investment options, and allows a better use of data innovations.

The inclusion of data innovations allows the evaluation of the portfolio with the arrival of the new data. In this case, the current state of wealth can be compared with the investor's expectations and take further actions if the result is undesirable. This work proposes decision rules called *Investment Strategies* (IS) to determine the optimal set of portfolios given the current state and the investor's expectations. The methodology to integrate evolutionary algorithms with investment strategies is explained in later chapters.

1.5 Main Contributions

The main contributions of this work are summarized as follows:

- This work presents a survey about the application of evolutionary algorithms to financial problems (Aguilar-Rivera, Valenzuela-Rendón, & Rodríguez-Ortiz, 2015).

- A novel multi-objective evolutionary algorithm is proposed to solve the problem. This algorithm proposes the called Area Measure A_{ms}, which favors individuals close to the non-dominated front located at low populated areas of the solutions space. This metric provides continuous fitness values, which differs from other found in the literature. Also, it implements a density measure designed to work when the algorithm is close to convergence.

- The proposed multi-objective algorithm is further modified to allow multi-objective stochastic optimization. Evaluation-saving methods are proposed for this end. This approach is able to find robust solutions without imposing a structure to uncertainty, which is a common approach in robust optimization.

- A set of suitable parameters is proposed to include the investor's preferences into the multi-period portfolio selection problem instead of using theoretical utility functions.

- This work shows the existence of multi-period efficient frontiers with heterogeneous time horizons, while current multi-period theory is focused on efficient frontiers with homogeneous time horizons.

- The investment strategies method is developed in this work. They use data innovations and Pareto fronts with heterogeneous time horizons to adapt decisions to market changes.

1.6 Thesis Organization

This chapter summarized the ideas to be developed in this work. Chapter 2 states the context through a review about the application of evolutionary algorithms to financial problems. Chapter 3 explains the new multi-objective algorithm proposed in this work, while chapter 4 explains further modifications to allow it perform stochastic optimization. Saving-evaluations methods are introduced at that moment. Chapter 5 explains the investment strategies method, the parameters to describe the investor's preference, and the existence of Pareto fronts with heterogeneous time horizons. They are integrated into a single investment method. Chapter 6 is a discussion about the data set used in the experiments and some considerations when working with real data from stock markets. Also, it presents a discussion about some metrics of portfolio performance. Chapter 7 presents the experiments and the discussion of results. Finally, chapter 8 presents the conclusions and the future work.

Chapter 2

State-of-the-Art

The literature review is presented in this chapter[a]. The first part reports some solutions to the multi-period portfolio selection problem. Evolutionary approaches are not mentioned in this section. Some of the references cited in past chapters are included here for completion purposes. The second part is a review about financial applications of evolutionary algorithms. The scope is limited to population-based approaches derived from genetic algorithms. The conclusions are found at the last section.

2.1 Multi-Period Portfolio Selection Review

Traditional portfolio theory is used by financial professionals in combination with fundamental or technical analysis to give investment advice. Nevertheless, the theory has limitations. For example, it assumes securities fractions are traded when only round-lots are usually allowed. Besides, it does not takes into account the impact of transactions in prices, which can be affected by changes in the supply and demand of securities. Also, portfolio theory assumes securities probability distributions are time-invariant. In practice, this is seldom the case (Hamilton, 1994). Also, real-world limitations such as transaction costs are overlooked. The investor can still choose a set of optimal portfolios, but the final wealth will differ from the calculations because transaction costs were not considered. Financial professionals usually select portfolios applying the traditional approach. Then, transaction costs are considered to the already optimized decisions (Sharpe et al., 1999).

There is no guarantee the optimal solutions before transaction costs are the same to the ones which considered them. Robust solutions frequently differ from non-robust solutions. Inflation is also a factor to be considered when making practical investment decisions.

The literature reports different efforts to deal with these limitations. The interest of the present work is focused on the effect of transaction costs and dynamic restrictions, which have received an important amount of theoretical study. For example, Fama (1970a) proposed a multi-period model considering the portfolio would be liquefied at time T. The decisions seek to optimize the expected value of return at time T instead of the present time. A solution was found using dynamic programming. The conclusion was there is no difference among risk-averse single-period strategies and multi-period strategies under the assumptions of no

[a]This chapter was edited to be published at an indexed journal (Aguilar-Rivera et al., 2015).

transaction costs and a perfect market. A. H. Chen et al. (1971) proposed an extension of the Markowitz's model to multiple periods to allow portfolio revisions. A single-step model is developed in that work. Transaction costs were considered at each step. The extension to a multiple-period model is developed using dynamic programming. Although, a solution is not shown for this case. Later, A. H. Chen, Jen, and Zionts (1972) studied single-period models which allowed stochastic cash demands from the portfolio. The conclusion was the presented model could be used to evaluate empirical studies.

Kamin (1975) presented an analysis for portfolio revision which considered transaction costs and risky assets. Constantinides (1979) also studied models where consumption is possible, but transaction costs are modeled with concave functions. An utility function is derived from the model which is proved to be state-dependent. The optimal set of portfolios from $t = 1, 2, \ldots, T$ (i.e the *policy*), is shown to be of no transactions as long as the portfolio weights lie inside a certain interval. These limits are function of the state of the world.

Elton and Gruber (1974a) stated single-period and multi-period problems are different when the investor utility function is state-dependent. Also, Elton and Gruber (1974b) studied multi-period problems of maximization of the geometric mean of future returns and future expected utility. The conclusion was both approaches lead to different optimal selections than multiple single-period decisions. This depends mainly on the utility function. Besides, continuous revision of portfolio leads to a higher utility when transaction costs are zero. Brennan (1975) presented a model to select the best stocks considering a mean-covariance model. He concluded the number of assets is dramatically reduced if the model considers transactions costs.

Barry and Winkler (1976) addressed the discrepancy of using static probability distribution models to describe non-stationary markets. A Bayesian model was proposed for security price forecasting. The conclusion was non-stationarity does not affect membership into the efficient frontier. Nevertheless, the efficient frontier is shifted to higher risk levels. Patel and Subrahmanyam (1982) proposed an algorithm to find the optimal portfolio with transaction costs in a single-period framework. This algorithm can be applied when pairwise correlation values of the securities are closely similar. Dumas and Luciano (1991) found an exact solution for dynamic portfolio selection with an infinite time horizon and transaction costs. This solution appears in the form of two control barriers where the portfolio is allowed to fluctuate. Mulvey and Vladimirou (1992) modeled multi-period problems and other financial problems using stochastic generalized networks (SGN). They concluded this representation could lead to parallel implementations of the solution. On the other hand, the size of the network grows rapidly with the number of periods and assets. Statistical methods and clustering analysis were suggested to alleviate this problem.

More recently, D. Li and Ng (2000) presented an optimal solution for a multi-period model with a mean-variance formulation. An algorithm was proposed to compute the optimal solution. Transaction costs are not considered. This approach was based on the definition of sub-problems to by-pass the difficulties of the original problem. Kellerer, Mansini, and Speranza (2000) remarked the one-period portfolio optimization with transaction costs and minimum lots is a NP-complete problem. They proposed an algorithm where heuristics and linear programming models are used to find the optimal solution. Steinbach (2001) proposed a model of multi-period problems based on scenario trees. Also, the possibility to use other risk measures besides variances was investigated in that reference.

Brandt and Santa-Clara (2006) proposed a multi-period solution which is implemented using quadratic programming. This algorithm optimizes sets of portfolios instead of single securities. Their hypothesis was the optimization of static sets is equivalent to a dynamic allocation of securities. Value-at-risk is used instead of the covariance matrix to measure risk. Value-at-risk is the q-quantile of the distribution. Value-at-risk is a common risk measure in financial frameworks. Transaction costs were not considered.

Some references have proposed multi-period optimization methods using Markov-chains (Çakmak & Özekici, 2006). Chiu and Li (2006) found an optimal solution for a mean-variance optimization problem with transaction costs and liability using stochastic linear-quadratic control, a technique borrowed from optimum control engineering. This work was extended by Liu, Zhao, and Zhao (2012), where a closed-form solution was found for the same problem, but assuming the stochastic process of prices satisfy jump-diffusion stochastic differential equations.

Z. Li, Yang, and Deng (2007) studied the effects of using earning-at-risk to multi-period problems. This measure can be understood as the $1 - q$-quantile of the distribution. A closed-form solution was found and compared with solutions derived from other risks measures.

Buy-and-hold is a simple strategy where the initial portfolio is unchanged until sold. Buy-and-hold has the advantage of minimal transaction costs, therefore, it is widely used by investors. Some references have compared multi-period re-balancing strategies against it (Xiong, Xu, & Xiao, 2009). The cases where one of the strategies outperforms the other one were explained in that reference.

Çanakoğlu and Özekici (2009) studied multi-period problems with exponential utility functions. Yi (2010) proposed a solution for multi-period models with transaction costs and a quadratic utility function. The solution is found by defining a set of auxiliary problems and solving them. Comparative studies between variance and conditional Value-at-risk were found (A. H. Chen, Fabozzi, & Huang, 2012). Both measures were used to model portfolio revision problems with transaction costs. This reference explained the way to integrate transaction costs to a conditional value-at-risk framework. Some analytical solutions were found under certain conditions.

Closed-form solutions for the multi-period problem have been found only for theoretical problems with particular properties. However, a closed-form solution is not necessary for a practical application (Desai et al., 2003). That work proposed an algorithm for a portfolio of stocks and one fixed-income asset using a Monte-Carlo approach. The example seems indicate numerical methods can be used to solve the problem in practical applications. Hibiki (2006) also proposed a Monte-Carlo approach. Besides, linear programming models were used to simulated paths while solving large-scale decision problems.

2.1.1 Multi-Period Literary Review Conclusion

This review showed the efforts found in the literature to expand the traditional portfolio approach and overcome its limitations. The multi-period definition of the problem allows the inclusion of transaction costs and other dynamic restrictions. It was concluded the solutions for single-period and multi-period problems are different when the utility function of the investor is state-dependent. The literature has focused to find closed-form solutions for the multi-period problem for a given set of conditions. Some of these works proposed a Monte-Carlo approach to numerically solve the problem. A Monte-Carlo method avoids the complications imposed by the dynamic restrictions of the problem.

The review concluded a Monte Carlo approach can be used to handle the multi-period portfolio selection problems. This work, besides, states this approach can be implemented with evolutionary algorithms. In this case, the objective function of the problem can be stochastic. Re-sampling could be used to estimate the probability distribution of final return. Besides, evolutionary algorithms allow the inclusion of transaction costs and other dynamic effects without further mathematical complications.

2.2 Evolutionary Algorithms and Financial Applications

There are three main reasons to use evolutionary algorithms in financial applications:

- The limited reasoning hypothesis (LRH) (Lakemeyer, 1994) is a realistic assumption to complete the efficient market hypothesis (EMH) (Finger & Wasserman, 2004). This means all investors are keen to make the best decision possible, but this one will depend on their computation power and ability to process financial data.
- There is a massive amount of financial data available like never before in history.
- The computational power available is vast and increases continuously.

The efficient market hypothesis holds it is impossible to attain extraordinary profit in a market where all the investors are rational and the market is efficient. There are many definitions of market efficiency, one of them is that all the possible information about the market is openly available to the investors at the same time. Rational investors will make the optimal decision under those conditions (Malkiel & Fama, 1970). On the other hand, the limited reasoning hypothesis holds some problems are too complex to process all the available information in a reasonable time to make optimal decisions. Therefore, the limited reasoning hypothesis does not contradict the rationality assumption, but realizes the investor might fail to foresight all the possible options available.

2.2.1 Genetic Algorithms and Darwinian Approaches

The number of approaches proposed for financial applications is vast. The scope of the survey must be delimited to find proper conclusions. Genetic algorithms (GAs) are included in the category of evolutionary algorithms. Nevertheless, this class includes other methods inspired by nature, culture, and language. Memetic algorithms are an example of culture-inspired evolutionary algorithms. Genetic algorithms are a simplified version of Darwinian evolution. Good solutions are favored to pass their distinctive traits to the next generation (Goldberg, 1989). This is possible by the use of genetic operators like selection, crossover, and mutation over encoded solutions.

Moreover, a survey on evolutionary algorithms imposes the problem to report comparisons between methods which are inherently different from each other. The fact the selected benchmark problem could favor a certain method is latent. Therefore, it is better to compare approaches with similar traits only. For example, the difficulty of genetic algorithms to solve the traveling salesman problem (TSP) was reported in early references (Grefenstette, Gopal, Rosmaita, & Van Gucht, 1985). On the contrary, other authors reported Simulated Annealing (SA) having good performance for problem sizes up to 200 cities (Johnson & McGeoch, 1997). At that time, the use of TSP instances as a benchmark could lead to a biased comparison between genetic algorithms and simulated annealing. It should be noticed that recent works have reported methods to apply genetic algorithms to large instances of TSP (Nguyen, Yoshihara, Yamamori, & Yasunaga, 2007). Approaches with a similar structure were chosen to avoid these problems.

Besides genetic algorithms, there are other methods based on Darwinian evolution. Some of them have found application in finance. The methods are listed and explained below:

- Genetic Programming (GP)
- Learning Classifier Systems (LCSs)
- Multi-Objective Evolutionary Algorithms (MOEAs)
- Co-evolutionary Optimization Schemes
- Estimation of Distribution Algorithms (EDAs)

Genetic programming was initially proposed by Koza (1992). It evolves solutions encoded as trees instead of binary strings. This allowed to find programs, mathematical proofs, electronic circuits, etc. Ferreira (2001) proposed gene expression programming (GEP). This format allows the use of binary strings to represent trees. Miller and Thomson (2000) presented Cartesian genetic programming (CGP), where programs are represented as indexed graphs. Ryan, Collins, and Neill (1998) proposed grammatical evolution (GE), which uses the Backus Naur Form to design a system to evolve high-level programs. This approach has the advantage to be language independent. Recently, some references presented Multi-stage Genetic Programming (MSGP) (Gandomi & Alavi, 2011). This method is used to model non-linear systems, considering the effect of individual prediction variables and its correlation with each other.

Learning classifier systems were proposed to find descriptions of changing environments (Holyoak & Holland, 1989). Further research proposed modifications to the original system to overcome its limitations. An example is the accuracy-based classifier system (XCS) (Wilson, 1994). Other authors also studied the XCS (Holmes, Lanzi, Stolzmann, & Wilson, 2002). The zeroth level classifier system (ZCS) was also proposed to overcome the limitations of learning classifier systems (Wilson, 1994). Valenzuela-Rendón (1991) designed the fuzzy classifier system (FCS). Stolzmann (2000) reported the anticipatory classifier system (ACS). Gerard, Stolzmann, and Sigaud (2002) proposed the latent learning classifier system (YACS).

Multi-objective evolutionary algorithms are used to perform Pareto optimization. In this approach, a set of solutions (the Pareto front) is obtained when two or more objectives are simultaneously optimized. The population is used to find a set of optimal solutions in parallel. The niched-Pareto genetic algorithm (NPGA) (Horn, Nafpliotis, & Goldberg, 1993), and the vector-evaluated genetic algorithm (VEGA) (Schaffer, 1985) were some of the earliest contributions. The Pareto archived evolutionary strategy (PAES) (J. D. Knowles & Corne, 2000), The non-dominated sorting genetic (NSGA-II) (Deb, Agrawal, Pratap, & Meyarivan, 2000), and a new version of the strength-Pareto evolutionary algorithm (SPEA-2) (Zitzler et al., 2001) were proposed in more recent reports. The multi-objective Bayesian optimization algorithm (MO-BOA) (Laumanns & Ocenasek, 2002) is a competent multi-objective algorithm; this definition of competence proposed by Sastry and Goldberg (2003).

Co-evolution was firstly presented by Hillis (1990). The concept of competitive co-evolution was used to evolve sorting networks in this work. Potter and De Jong (1994) used a cooperative genetic algorithm (CCGA) for function optimization. This algorithm showed a better performance than a traditional genetic algorithm in the experiments. Chang (2010) applied co-evolution to solve supply chain network design problems.

Estimation of distribution algorithms (EDAs) are mainly related to the estimation of population probability distribution. They do not rely directly on genetic operators. They are presented in this review because their process is an abstraction of Darwinian evolution and genetic operators. Harik, Lobo, and Goldberg (1999) proposed the compact genetic algorithm to model the population with a probability distribution; it was designed to emulate a traditional genetic algorithm performance. The extended compact genetic algorithm (ECGA) was explicitly designed to solve problems where linkage was present (Harik, Lobo, & Sastry, 2006). Pelikan, Goldberg, and Cantú-Paz (2000a) reported the Bayesian optimization algorithm (BOA), where Bayesian networks were used to model the non-linearity of the objective function. There is also an extension designed to solve hierarchical problems (Pelikan, Goldberg, & Cantú-Paz, 2000b).

2.2.2 Refining the Scope of the Review

The existence of surveys with similar approaches to this review seemed plausible. Therefore, there is the possibility to cover references already cited in similar works. For this reason, the scope was delimited by the review of surveys with related subjects than this one. The novelty of this approach comes in three different manners: This work covers lapses not included in other review articles, it covers problems not considered by others, and the scope covered by past references and new ones is compared and analyzed.

A review to find similar surveys was conducted from 2003 to 2015. Table 2.1 presents a list of articles with similar scopes. The problems considered are now enlisted: Abnormal noise (fraud) detection (ABN), arbitrage (ARB), bankruptcy detection (BKR), cash management (CM), credit portfolio (CP), credit scoring (CS), fundamental analysis (FA), forecasting (FC), index tracking (ITR), market simulation (MKS), procurement (PCR), portfolio selection problem optimization (PSP), trading (T), and trading execution (TX). These problems are further explained in the following section.

The reviews considered were the following ones: (S.-H. Chen, 2003), (Vanstone & Tan, 2003), (E. P. Tsang & Martinez-Jaramillo, 2004), (Tapia & Coello, 2007), (Atsalakis & Valavanis, 2009), (Hruschka, Campello, Freitas, & De Carvalho, 2009), (Bahrammirzaee, 2010), (Lahsasna, Ainon, & Teh, 2010), (Phua, Lee, Smith, & Gayler, 2010), (Safarzyńska & van den Bergh, 2010), (Verikas, Kalsyte, Bacauskiene, & Gelzinis, 2010), (Ngai, Hu, Wong, Chen, & Sun, 2011), (Ponsich, Jaimes, & Coello, 2013), and (Giulioni, D'Orazio, Bucciarelli, & Silvestri, 2015).

This review concluded none of these works covers the exact approach of the present report. Some of them are mainly focused on other areas instead of evolutionary algorithms.

Reference	Techniques	Problems
(S.-H. Chen, 2003)	GA,GP	MKS
(Vanstone & Tan, 2003)	GA	T,FA
(E. P. Tsang & Martinez-Jaramillo, 2004)	GA,GP,LCS	MKS,FC
(Tapia & Coello, 2007)	GA,MO	FA,FC,MKS,PSP
(Atsalakis & Valavanis, 2009)	–	FC
(Hruschka et al., 2009)	–	CS,BKR
(Bahrammirzaee, 2010)	–	CS,FA,FC,PSP
(Lahsasna et al., 2010)	GA,GP	CS
(Phua et al., 2010)	GA,GP	ABN
(Safarzyńska & van den Bergh, 2010)	GA,GP,LCS	MKS
(Verikas et al., 2010)	GA	BKR,CS
(Ngai et al., 2011)	–	ABN
(Ponsich et al., 2013)	GA,GP,MO	BKR,CP,FA,FC,PSP,T
(Giulioni et al., 2015)	GA,LCS	MKS

Table 2.1: Summary of Review Articles

Some of these works do not consider any of the techniques reviewed in this work. The exact contents in these works was further investigated. Table 2.2 shows the span in years covered by the references for each solution method. Specific time ranges are provided because the coverage of a specific solution method by a reference is not always equal to the total time range covered by it.

This analysis concluded co-evolution and EDAs are the techniques less used for financial applications. At least, they are not directly referenced by other review articles. They do not appear in table 2.2 for this reason. Genetic algorithms and genetic programming seem to be the most studied methods, while learning classifier systems have attained limited attention. Problems like arbitrage (ARB), cash management (CM), index tracking (ITR), procurement (PCR), and trading execution (TX) were not reported by other review articles, but references about them were found in the review conducted for the present survey. References with overlapping dates to these time ranges were eliminated from this study. The following section presents the review of uncovered references.

2.2.3 Description of Financial Applications

This section makes a description of each application found in the literature. The review is classified by problem.

Abnormal Noise and Fraud Detection (ABN)

A recent application of genetic algorithms to financial environments is abnormal noise detection in markets. Abnormal noise is believed to be correlated with illegal practices like rat trading or money laundering (Jing, 2010). This reference described a genetic algorithm approach designed to detect abnormal noise. Although, experimental results were not presented

Problem	GA	GP	LCS	MO
ABN	1999	2000	–	–
ARB	–	–	–	–
BKR	1992 – 2011	2009 – 2010	–	2010
CM	–	–	–	–
CP	–	–	–	2002 – 2010
CS	1997 – 2007	2005 – 2006	–	–
FA	2001	2009	1998	1998
FC	–	1994 – 2004	–	1994 – 2009
ITR	–	–	–	–
MKS	1994 – 2013	1990 – 2003	1997 – 2006	1998 – 2011
PRC	–	–	–	–
PSP	1993 – 2011	–	–	1994 – 2010
T	1994 – 2002	1999 – 2006	–	2009 – 2011
TX	–	–	–	–

Table 2.2: Time Range per Technique Covered by Other Review Articles

in this work. Jun and Lei (2012) used a combination of genetic algorithms with neural networks (NNs) for money laundering detection. Genetic algorithms were used to modify the weights of neural networks only. The possibility of modifying the neural networks structure was suggested as future work.

Arbitrage (ARB)

Another problem of interest was arbitrage. Arbitrage is the practice to take advantage of price differences of the same asset. This occurs when information is not simultaneously updated to all the investors. Arbitrage is more common among stocks from different markets. Markose, Tsang, Er, and Salhi (2001) proposed a genetic programming algorithm to find arbitrage opportunities for the London stock index (FSTE-100) futures and options. This algorithm was trained with historical data where put-call-future parity cases are detected. Genetic programming is used to make an estimation of the longest time the arbitrage will remain profitable. The estimation is used to make earlier transactions than the traditional method where a contemporary signal of profitability is required. Financial GP-2 was developed to be an interactive tool to find arbitrage opportunities using intra-daily data. This approach attained greater returns than the textbook approach.

E. P. Tsang et al. (2000) designed the evolutionary dynamic data investment evaluator (EDDIE), which was later used to for arbitrage applications (E. Tsang, Markose, Garcia, & Er, 2006). EDDIE relies on a human expert to evaluate the relevance of information. The results are later used by the system for the optimization process. EDDIE has proved to process financial information more efficiently than a human expert alone. An online implementation was later reported (E. Tsang, Yung, & Li, 2004).

C.-F. Huang, Hsu, Chen, Chang, and Li (2015) proposed a genetic algorithms approach for pair trading. Pair of stocks are bought and sold combined to find arbitrage opportunities. Trading pairs is based on the idea that mispriced assets will eventually reach their true value. The trading pairs method chooses a pair of stocks from the same industry, sells the one with relatively high price and buys the one with relatively low price. These stocks are likely to be mispriced and to converge to their real prices in the future. The method closes position to attain profit when the spread has reduced to a certain threshold value. The proposed method considered moving averages (MAs) and Bollinger bands to estimate the long term mean of stocks and thresholds, respectively. Genetic algorithms are used to optimize these parameters, besides portfolio weights of stock pairs. The fitness value of individuals is the annualized return obtained from those specific trading pairs. The method was tested using 10 stocks from the semiconductor industry of the Taiwan stock exchange and 10 stocks with the highest capitalization from the same index. The benchmark is a equal-weighted buy-and-hold portfolio of the mentioned stocks. The proposed method outperformed the benchmark in both experiments.

Bankruptcy Detection (BKR)

Bankruptcy detection is the task of determining beforehand if a company is prone to bankruptcy in the near future. Accounting information and financial data are used to estimate the likelihood of the worst case.

Varetto (1998) made a comparison between genetic algorithms and traditional statistical methods for bankruptcy detection. Accounting information and financial reports were used to determine a company's health. Genetic algorithms were able to obtain comparatively good results with less data than the statistical approach in the experiments.

Multi-objective algorithms have also been proposed for bankruptcy detection applications (Gaspar-Cunha, Recio, Costa, & Estébanez, 2014). When companies grow they become complex and the estimation of their real financial situation becomes difficult. Financial reports of complex companies have a large number of features, and many of them are irrelevant to estimate their actual status. That reference proposed a multi-objective evolutionary algorithm to minimize features and maximize the accuracy of classifiers. A feature selection algorithm is applied periodically while parameters are optimized. Each individual encodes the relevant features and the configuration parameters of the selection features algorithm, which is implemented using support vector machines (SVMs). Support vector machines are applied to training data and the accuracy of the specific configuration is used as the fitness value. Its accuracy is computed from the confusion matrix of the test. The individuals are ranked according to their fitness value, and a fixed number of the best individuals are copied to a secondary population, which is larger then the primary population. The secondary population is completed with offspring generated from the selected individuals. A fraction of the best individuals from the secondary population substitute the worst individuals of the primary population. The method was tested with data from industrial French companies (DIANE database), Australian credit data (UCI Machine Learning Repository), and German credit data (UCI Machine Learning Repository). The results showed that the algorithm was able to significantly reduce the number of features, but better accuracy was obtained with larger sets. Besides, the multi-objective approach provided a set of solutions with different levels of simplicity and accuracy. Inclusion of the decision maker preferences to obtain a single solution was indicated as future work.

Cash Management (CM)

Some references have studied the problem of cash management (da Costa Moraes & Nagano, 2014). Cash management is related with the accumulation and use of cash in companies and institutions; companies need cash for their operations, and a lack of liquidity will obstruct procurement and direct investment. For example, mutual funds need to plan how to invest their costumers money to provide the promised returns while providing daily liquidity. Precaution and speculation are other reasons to maintain a cash balance.

The approach proposed by da Costa Moraes and Nagano (2014) considered a stochastic model where current cash will vary along a bounded range. Genetic algorithms and particle-swarm optimization (PSO) were used to find the optimal strategy. The method considered a portfolio composed of two risky assets and cash. The system should decide the amount to be invested or saved to satisfy possible cash flows. A Miller-Orr model for full cash management was implemented (i.e., cash is stochastic). Nevertheless, cash bounds were added to the original model. Cash flows were assumed to be a random variable with normal distribution. Transaction costs, interest rates for lending, bounds form risky assets, and asset liquidity are the parameters of the model. Genetic algorithms and particle-swarms searched to minimize cash management costs in a multiple-period time horizon. Different probability distribution configurations were used to simulate cash flow in the experiments. Both algorithms were capable to determine cash flow policies, but Genetic algorithms outperformed particle-swarms for most of the instances tested. Nevertheless, particle-swarms attained lower average relative deviation than genetic algorithms. This work concluded both techniques are viable methods to further study the problem.

Credit Portfolios (CP)

Credit portfolios are similar to security portfolios, but they are composed of credits instead of securities. Banks should decide which credits to approve to maximize profit and minimize risk. The profit comes in the form of interest, while losses occur when the borrower fails to pay the debt back. This problem was mentioned in other reviews (Tapia & Coello, 2007) and (Ponsich et al., 2013). Both references proposed multi-objective algorithms to solve the problem.

Credit Scoring (CS)

Credit scoring is the problem of determining if applicants will repay their loans to banks. Credit approval is made based on this estimation. Mis-classified applicants represent a loss for banks and loaners. Banks and stores invest time and resources to select the most promising applicants.

Back, Laitinen, and Sere (1996) proposed a combination of neural networks and genetic algorithms to perform this task. Genetic algorithms were used to select the input variables for the neural networks. The results showed the approach was successful for the experiments presented. It was concluded the prediction was only reliable for a one year time window. Ravi, Kurniawan, Thai, and Kumar (2008) proposed a sophisticated approach which combined neural networks, support vector machines, decision trees and genetic algorithms to solve this problem. Genetic algorithms were used to optimize the weights of the neural network. The approach showed to have less Type-I (false positive) and Type-II (false negative) errors than the individual techniques. It also performed better than a neural network trained with human expert data.

Hochreiter and Wozabal (2010) used a coupled Markov-chain approach to compute the fail probability of credit scoring. In this work, maximum likelihood model estimators were found using genetic algorithms. Neural networks trained with genetic algorithms has also been combined with regression methods for this application (Nikolaos & Iordanis, 2010). The proposed method showed a higher rate of correct decisions, but its minimum squared error was higher than the error of the regression methods. The applications of different neural network structures was proposed in the future work section. Other examples of the combination of support vector machines and genetic algorithms were found in the literature (Lin, Liang, Yeh, & Huang, 2014). In the experiments, this approach showed better results than the traditional analysis of discriminant method. The future work section suggested the approach needs a method to define the algorithm's parameters.

Fundamental Analysis (FA)

Fundamental analysis is concerned with the valuation of securities. It makes use of the company's accounting and financial information for this end. A careful estimation allows profitable opportunities to be found. This occurs when an asset is over-valuated or sub-valuated at the market. The investor can take advantage of this information before assets reach their true value.

Jiang, Xu, Wang, and Wang (2009) proposed the use of genetic algorithms for this end. A genetic algorithm selected the most significant variables to determine the company value. The selected features were analyzed using discriminant analysis to estimate future financial performance of Chinese companies. It was concluded that the approach was effective to find the best selection; this selection changed accordingly with the company line of business.

The information of initial pricing offerings (IPOs) can be useful to fundamental analysis applications (C.-F. Huang et al., 2012). An IPO contains the information used to determine the asset value when it was initially introduced to the market. The approach used this data to rank securities. A genetic algorithm was used to determine the most significant indicators, besides the most suitable weights for each one of them. The objective is to find an equal-weighted portfolio with the maximum return based on these indicators. The rules can be applied later to different securities. This fundamental analysis approach attained positive returns in the experiments, which means the selected stocks attained higher first-day return than the average of the whole set of securities.

Forecasting (FC)

The forecasting problem is one extensively studied. It consists in the estimation of future values of securities and trends in data. Investment would be straightforward with a perfect predictions of the future. Although this is not possible, forecasting involves also an estimation of the prediction error, allowing to make better decisions under uncertainty.

Packard (1990) presented one of the earliest authors to use genetic algorithms to solve this task. Also, the same approach has been applied to find prediction rules for the Mackey-Glass equation, which shows chaotic behavior (Meyer & Packard, 1992). Kingdon, Taylor, and Mannion (1997) presented applications of genetic algorithms and neural networks to forecast financial time-series. The application of genetic algorithms for option pricing has been reported in the literature (S.-H. Chen & Lee, 1997). This approach was tested with the European call Option problem, which exact solution is known. The experiments showed positive results, suggesting a practical implementation is attainable.

Kim and Han (2000) proposed the use of a neural networks for the prediction of price indexes. The continuous variables were mapped into discrete sets. A genetic algorithm was used to find the optimal ranges for the input variables. Mathematical transformation approaches were also found (Ma, Wong, Sankar, & Siu, 2004). This reference reported the use of wavelet transform and genetic algorithms to forecast the volatility of financial indexes. The obtained coefficients were processed with genetic algorithms to find useful patterns. The approach was tested against GARCH models and attained positive results.

Rimcharoen, Sutivong, and Chongstitvatana (2005) proposed an $(\mu + \lambda)$-ES (evolutionary strategy) to solve the problem. It was tested with data from the Thailand stock market. This approach showed a better performance than multiple regression. The results showed the existence of a high correlation between the Thailand and the Taiwan stock markets, which was exploited by the algorithm.

Polanski (2011) reported multi-dimensional time-series forecasting is also an interesting problem. This work presented an experiment with foreign exchange market (FOREX) data. de Brito and Oliveira (2012) compared the performance of genetic algorithms with a hybrid SVM-SOM (Self Organized Maps) for FOREX trading. The genetic algorithm attained higher returns in financial crisis scenarios.

The application of fuzzy theory to forecasting problems was also found in the literature (Goonatilake, Campbell, & Ahmad, 1995). That work proposed the use of fuzzy systems to make trading decisions. A genetic algorithm was used to find the optimal fuzzy sets configuration. The hybrid system generated meaningful rules for human experts. Kanungo (2004) proposed a combination of genetic algorithms with maximum likelihood estimation. Although the good results, it was concluded more experiments were necessary to prove its reliability. Modular morphological neural networks (MMNNs) and genetic algorithms have also been applied to solve the forecasting problems (de Araujo, Madeiro, de Sousa, Pessoa, & Ferreira, 2006). The approach was tested with data from the Standard & Poor (S&P500) index. The experiments concluded the algorithm seemed to model time-series as a random walk. The inclusion of a correction module was suggested as future work.

Parracho, Neves, and Horta (2011) used genetic algorithms to find trend patterns and use them to predict market tendencies; it was proved with data from $S\&P500$ index. The method showed positive results for the experiments presented. Araújo and Ferreira (2013) used genetic algorithms to optimize linear filters in forecasting applications. The obtained filters were further enhanced using minimum squared-error estimators. The approach was compared against neural networks with time-delay evolutionary forecasting (TAEF), and random walk models. The proposed approach outperformed the benchmarks in the experiments. Further research on the method properties was suggested as future work. Bernardo, Hagras, and Tsang (2013) used a type-II fuzzy system adjusted with genetic algorithms for this task. This approach performed better than genetic programming. The performance was similar to neural networks, but it has the advantage of producing results which are meaningful to human experts.

Ghosh and Chinthalapati (2014) proposed agent-based modeling to forecasting of financial markets. A non-equilibrium economics was considered and their features were included into a set of bounded-rational and heterogeneous agents. Interaction was possible between agents, and genetic algorithms were used to model their behavior at an artificial market. Time series were binary, this means they only represent the current directional trend of prices. N agents populated the economy and each one had a number of strategies ranked according with their performance and limited memory of the past. Agents should decide to buy or sell based on current state and their own strategies. Four types of agents were considered: Minority game, majority game, $-game, and delayed minority agents. Minority agents are rewarded when their decision is opposite to the actual market. Majority agents are rewarded when their decisions are the same as the actual market. $-game agents assume last market value is the best estimation of future and are rewarded when this occurred. Delayed agents make the opposite assumption than $-game agents. A genetic algorithm with islands is used to simulate the market. Each island is a sub-population which represents a possible optimal market. Each island is populated with agents with heterogeneous beliefs. Fitness of each individual depends of the agent type they represent. Interaction among islands is possible, but each one follows their own optimization process. The experiments presented two cases: FTSE-100 closing prices and FOREX case. Binary time series are obtained from original time-series of prices. The market returns were the ones evaluated instead of individual performance. This means islands are evaluated instead of individual agents. It is possible to obtain a forecast of future price based on the returns of each island. The forecasts were used to implement a trading strategy based on the obtained market models. Agents obtained a hit ratios in out-of-sample data of about 67%. The authors suggested using the universal information criterion in future work to better study the significance of the obtained results.

Garcia-Almanza and Tsang (2006) studied the detection of bubbles and crashes in financial markets. These events are hard to predict because they rarely occurs, nevertheless, they have heavy impact in markets. The repository method (RM) is an analysis technique which was applied to the decision trees generated by genetic programming. This method extracts and simplifies rules encoded in each individual, adding them to a repository if they cover different possibilities than current ones. The work presented experiments designed to discover the factors required for the good performance the repository method. It was concluded that the accumulation of rules is crucial to classify correctly the positive cases.

Wagner, Michalewicz, Khouja, and McGregor (2007) proposed a modification to traditional genetic programming to deal with dynamic environments. This was called dynamic forecasting genetic programming (DyFor GP). It includes an adaptive sliding-window approach chooses the best window size to describe the environment. It compares different windows sizes and chooses the one which allowed the best prediction of current data. This one is used to forecast the next future value. It was applied to estimate U.S. Gross Domestic Product. DyFor GP results were better than regular genetic programming and other benchmarks models.

The financial forecasting tool EDDIE mentioned above has also been applied to forecasting (Shao, Smonou, Kampouridis, & Tsang, 2014). This new version provided an extended grammar for the generated decision trees. Although, this new grammar increased the search space. This work proposed a combination of guided local search with fast local search (GFLS) to solve this problem. Technical indicators were used as inputs. Both the rules and the forecasting time horizons were optimized. Only the latter are subject to GFLS. GFLS uses hill climbing to improve current solutions. Guided search modifies hill climbing behavior through solution classification, fitness modification and class penalization. Fast search divides the space in sub-neighborhoods and eliminates those where no improvement was found to save computational effort. The method was tested using data from different market indexes like FTSE-100, Dow Jones industrial average, Nikkei-225 and others. The results indicated the new approach improved the performance of EDDIE when comparing it to GLS alone. GLS is the same algorithm without the effort-saving capabilities.

Hamida, Abdelmalek, and Abid (2014) proposed a genetic programming algorithm for volatility forecasting. Volatility is the implied variance of return at a given time. Volatility can be estimated from historic data or from observed option prices. Option prices show the expectation about the future price that the underlying asset will reach at maturity time. Forecasting rules are optimized using genetic programming, where both the historic and option data are used as inputs. One of the problems to be solved by this approach is determining the sample size used to compute estimations. Four methods to determine sample size were proposed in this work: random subset selection (RSS), sequential subset selection (SSS), adaptive random subset selection (ARSS), and adaptive sequential subset selection (ASSS). The method used was applied each g cycles of the algorithm. Random subset selection selects a sample size randomly with uniform distribution. Sequential subset selection applies each sample size using a predetermined sequence. Adaptive methods compute the average mean squared error obtained during g cycles for the sample size used. Sequence order is rearranged according with this measure for the adaptive sequential method. Selection probability is tuned in a similar manner for the sample sizes using the adaptive random subset selection method.

Karatahansopoulos, Sermpinis, Laws, and Dunis (2014) studied two different genetic programming approaches for forecasting applications. One was regular genetic programming, which evolves mathematical expressions represented as trees. The second one is gene expression programming. This work proposed a one-day ahead forecasting application to model the ASE-20 Greek index. Both techniques where tested separately against neural networks, ARMA models, MA models, and a naïve strategy where the current return value is taken to be the estimation of future return. Lagged index values and moving averages of the values from ASE index were the inputs for the proposed techniques. The proposed methods outperformed the benchmarks. Gene expression programming showed better annualized return than regular genetic programming.

Mahfoud and Mani (1996) proposed a genetic algorithm forecasting of security prices. The genetic algorithm was similar to the one included in a learning classifier system based on the Michigan approach. This kind of system encodes a rule per individual. A similar approach studied a non-generational Michigan genetic algorithm for forecasting applications (del Arco-Calderón, Vinuela, & Castro, 2004). Each individual tries to predict a particular case of the series. It was applied to predict random securities selected form S&P-400. This method showed the ability to detect regions which cannot be generalized. Both methods are closer to a learning classifier system than to traditional genetic algorithm. Credit assignment and conflict resolution are discussed in that work, which are important issues to solve when implementing learning classifier systems.

Donate and Cortez (2014) proposed a neural networks approach for forecasting applications. The difficulty of this approach is finding the best neural network model for the task. The uni-variate marginal distribution algorithm (UMDA) is used to search for this end. This one is classified under the category of estimation of distribution algorithms. This algorithm copies the best half of the current population into the new one; the rest of the individuals are randomly generated using the probability distribution computed by the method. Two design strategies for neural networks are considered: Sparsely connected neural networks and time lag selection neural networks. The former considers a binary direct encoding of neural networks which can be directly mapped into a matrix of connections. The latter considers a time-lagged feed-forward neural network; therefore, the lags are also searched by the algorithm. Neural networks are first optimized using resilient propagation before applying UMDA. Dow Jones index data is one of the time series used to test the approach. Comparison methods used were ARIMA models, random forest (RF), echo state networks (ESN), and support vector machines. The proposed method showed to attain lower mean-squared error the the other ones.

Index Tracking (ITR)

Market Indexes are extensively used as benchmarks in financial applications. They are also used as indicators of the health of economies. Different organizations take the most representative securities in the market to build their indexes. The Dow Jones Industrial Average (DJI), the Mexican Índice de Precios y Cotizaciones (IPC), the Japanese Nikkei 225, and the British FTSE-100 are examples of market indexes. Nevertheless, to build a portfolio with the exact composition of a market index is a difficult task. For example, round lots restrictions will make feasible replications to have a prohibitively high value. For example, a portfolio based on the FTSE-100 would require to buy all these securities. Information about the exact composition of IPC is quarterly published only (*Notas Sobre Índices*, 2015, February), which difficult its replication.

Index trackers are used to replicate the values of market indexes. The value of these instruments are usually a fraction of the true index value. Index trackers are not limited to the original index composition. Other instruments besides securities are used to build trackers.

G. Chen and Chen (2011) proposed an adaptive genetic algorithm for pattern recognition (AGA-PS) for index tracking purposes. This approach is a type of local search based on neighborhoods. Crossover and mutation probabilities are updated online. The authors selected 20 securities from Hushen-300 stock index and tried to replicate its value with minimum error. The approach showed better results than a traditional genetic algorithm for the same problem.

Andriosopoulos and Nomikos (2014) studied tracking of a spot energy index built from information from the New York mercantile exchange. Commodities futures have become a mean to attain effective investment diversification. Nevertheless, these instruments oblige their holders to provide the commodity traded at maturity time. This work proposed to invest in commodity-related equities instead to avoid this problem. The hypothesis is that a careful selection of securities from commodity-related companies is a suitable tracker of real commodities. This work proposed to solve using a differential evolution algorithm (DEA) and genetic algorithms. The average and the standard deviation of the tracking error are combined to compute a single-objective fitness value.

Securities from the Dow Jones, the Bovespa composite, and the FTSE-100 were used to build the tracker. Different number of stocks and different values of the trade-off parameter λ were tested in the experiments. The method is applied in buy-and-hold, quarterly, and monthly re-balancing scenarios. The results showed the method is capable of tracking the index. The Bovespa pool even outperformed the benchmark. The authors concluded that both equities and commodities have similar return distributions, making their tracking possible. 15 stocks and $\lambda = 0.8$ were the best combination of parameters.

Market Simulation (MKS)

Some studies have exploited the advantage of computational methods to investigate the properties of markets. Genetic algorithms and other techniques allow the simulation of agents with different behaviors. These agents are allowed to interact in artificial economies to observe the effect of their behavior on the economy. These studies are concerned with the validation of theoretical models. Agents which are built under the model assumptions are expected to attain the results predicted by it. On the other hand, agents with heterogeneous behaviors can be used to find more realistic models of the economy.

Kampouridis, Chen, and Tsang (2012) used genetic programming combined with self-organized maps (SOM) to simulate agents in a constantly changing economy. The hypothesis that a constant evolution of strategies is needed for the agents survival is concluded from the experiments.

Game theory concepts are a useful tool to model market behavior (Sinha, Malo, Frantsev, & Deb, 2014). That reference proposed a study about multi-period, multi-leader-follower Stackelberg game to model oligopoly economies. They are based on Cournot games. Cournot games consider an economy where many companies compete with each other to sell the same type of product. Each agent can influence supply and market price with their own production. They should estimate future price and decide their production based on their estimations to maximize profit. Stackelberg games differ form Cournot games because the former ones have two types of agents: leaders and followers. Leaders move first and posses the necessary information about followers to estimate their future actions. Followers observe the leader's actions and make decisions conditioned to them. This is a case of bi-level optimization because both leaders and followers are part of a Cournot game among their equals. This work proposed a model and applied it to an aircraft manufacturing industry case. A steady state, real-coded genetic algorithm is proposed to solve the model, making decisions for a number of periods ahead. The genetic algorithm works first the leader part of the chromosome. Parent centered crossover (PCX) and polynomial mutation were used for this end. The closest individuals to the offspring are identified and their follower part of the chromosome is copied into them. $n - 1$ random low-level individuals are generated to form a sub-population along with the copied information. New low-level individuals are generated using crossover. The low-level individual with less violations to restrictions is finally chosen. The algorithm stops when the average normalized population variance is smaller than η. The algorithm was capable of solving the instances presented in the experiments. Parallelizations were proposed in the future work section.

Co-evolutionary approaches were also reported in the literature (Franke, 1998). That reference proposed a co-evolutionary genetic algorithm scheme to simulate a cobweb economy with heterogeneous beliefs. A cobweb model explains prices fluctuation based on the expectations of producers about the future demand of their product. Producers should plan production based on their expectations. Changes in their expectations affect supply along with price. The modified genetic algorithm is similar to the one found in learning classifier systems. Martinez-Jaramillo and Tsang (2009) simulated a market populated with fundamental-analysis-based traders, technical-analysis-based traders, and noise traders. Technical agents used a co-evolution genetic programming approach to find their strategies. Both Single Population and Multiple Population models were studied. Single population models consider each individual as an independent agent. On the other hand, multiple population models use a whole population to represent a single agent. The red queen principle (RQP) is included in the model by taking forecasting precision as the fitness measure. The red queen principle holds constant evolution is observed in competitive environments. In this case, individuals must improve constantly, lest they will be left behind by new individuals with better accuracy. This work concludes heterogeneity, learning, and the red queen principle are factors that should be present in real markets.

Protopapas, Battaglia, and Kosmatopoulos (2010) used co-evolutionary genetic algorithms to simulate agents in Cournot games. They can affect prices with their own production. They should decide their production to maximize profit. The work concludes the games converges to Nash equilibrium in social learning scenarios. In these scenarios the information of all the populations is used together to update their strategies.

Procurement(PRC)

Procurement is the systematic process used by companies to purchase their necessary goods and services. Procurement is important to ensure purchases satisfy their requirements at the lowest price possible.

Some references treated this problem as a case of stochastic Programming (Tezuka, Munetomo, & Akama, 2007). They proposed modifications to a regular genetic algorithm to accomplish this task. This algorithm is applied to noisy objective functions, where the mean of individual fitness is estimated using a Monte-Carlo approach. The approach tries to determine the sample size of a pair of individuals which need to be discriminated using tournament selection. An F-test based method is proposed to estimate the sample size to minimize the variance of the average fitness of individuals. Bootstrapping is applied to avoid excessive evaluation. The experiments studied a case of procurement planning where a company should decide the amount of materials and time of purchase based on estimations of the market price of products. Market prices were treated as stochastic variables.

Portfolio Optimization (PSP)

Portfolio optimization is based on the concept of investment diversification. Modern portfolio theory (MPT) considers each investment decision implies risk, which is a measure of the possible loss investors could face when they make a particular decision. This theory proves a set of assets (i.e. portfolio) can attain less risk for the same expected return. Portfolio optimization searches these optimal portfolios, which is a case of Pareto optimization.

Modern portfolio theory proposes a quadratic optimization algorithm which solves efficiently a mean-variance description of portfolios. Nevertheless, the problem turns to be difficult when real-world restrictions are considered. Transaction costs, round-lots, composition boundaries, and non-stationary time-series are examples of these restrictions.

Gupta, Mehlawat, and Mittal (2012) proposed a hybrid method based on genetic algorithms and support vector machines. The first part applies a support vector machine to classify assets based on selected financial indicators: Liquidity, high return and low risk. The second part applies a real-coded genetic algorithm to build the portfolio. The investor's preferences are included when selecting one of the sets of securities determined by support vector machines. An optimal portfolio is searched using a weighted sum of these financial indicators. Boundaries of portfolio composition are considered into the process.

Wang, Hu, and Dong (2014) proposed a portfolio optimization model based on a convex risk measure called weighted expected shortfall (WES). In this measure, the cumulative probability of final portfolio value (which should be less than x_α) is weighted by an exponential function to compute WES. A coefficient λ is included in its argument to control risk aversion. The proposed model is a case of nonlinear optimization. Genetic algorithms were chosen to optimize the model. WES is the fitness value to be minimized. 10 random stocks from the Shenzhen index were randomly chosen. Portfolios optimized with different values of risk aversion coefficient were compared in the experiments. The optimized portfolios were kept in buy-and-hold for 60 days to test their performance.

Some references reported combinations of genetic algorithms with local search algorithms (Hochreiter, 2014). They proposed a genetic algorithm combined with local search to solve a risk parity portfolio selection problem. Risk parity portfolios are those where each stock weight is adjusted in a way each asset contributes equally to the total risk of the portfolio. The problem is trivial when long-only portfolios are allowed, but it turns difficult when short positions are possible. The objective functions is the sum of differences of average risk per stock. Elitist selection, and random addition of individuals are some of the specific properties of this algorithm. The best solution found by the genetic algorithm is then optimized using local search. The method was tested with data from the Dow Jones index the and S&P-100 index. It was compared against a minimum variance portfolio and an equal-weighted portfolio for the long-case. The long-short case was compared against randomly generated portfolios. The proposed method obtained the best results.

Genetic programming has also been studied to solve portfolio selection problems with different performance measures (Wagman, 2003). That reference proposed a genetic programming algorithm to find rules based on technical analysis indicators to design portfolios with high return of investment (ROI) rate. The algorithm searched for portfolios with higher ROI than the current market interest rate under conservative market conditions. The average from past prices, historic minimum, and maximum historic data were used as inputs to the system. Data from the DJI from 1979–1980 were used in the experiments. Future work involved considering capital adequacy into the optimization process. In other references, Krink and Paterlini (2011) used a differential evolution multi-objective approach for a similar application.

Static restrictions to portfolios are also of interest of some references (Lwin, Qu, & Kendall, 2014). That reference studied a mean-variance portfolio selection problem with cardinality, quantity, pre-assignment, and round-lots restrictions. Cardinality restricts the number of stocks in the portfolio. Quantity refers minimum and maximum proportion of assets in the portfolio. Pre-assignment restriction forces the algorithm to include certain stocks in the portfolio. Lots of a specific number of stocks can be traded only. This work proposed a new multi-objective algorithm for the task, and compared it against four popular multi-objective evolutionary algorithms: The non-dominated sorting genetic algorithm (NSGA-II), the Pareto envelope selection algorithm (PESA-II), the strength Pareto evolutionary algorithm (SPEA-2), and the Pareto archived evolutionary strategy (PAES). The proposed algorithm encodes portfolios using two vectors: One to indicate if the specific asset is part of the portfolio and another for the portfolio weights. The composition of non-dominated portfolios is observed; each one of their assets are given a concentration value proportional to their number of occurrences. Candidate assets are selected using their concentration values. Pre-assigned stocks are compulsory included. Three random portfolio are chosen from the population. Candidate portfolios are generated from these ones using mutation operators or scaling factors. Portfolio weights of individuals are determined selecting one of these methods randomly. The resulting weights are modified to comply with quantity and round-lots restrictions. The methods were compared using typical multi-objective measures like the Δ metric or the hyper-volume of solutions. The proposed method showed better performance than the benchmarks in the experiments.

García, Quintana, Galván, and Isasi (2014) studied the effect of re-sampling in multi-objective algorithms. This work solved a mean-variance portfolio selection problem with cardinality and quantity restrictions. Re-sampling was implemented using a bootstrapping method where a sliding window is used to determine the sample data; expected return of assets and covariances matrix were recomputed from this new sample. Popular multi-objective algorithms were modified to implement re-sampling. Data from the Frank Russel indexes and the Standard & Poor were used for the experiments. SPEA-2 attained the maximum improvement in quality solution by the inclusion of re-sampling.

Portfolio selection can be understood as a stochastic optimization problem (Hochreiter & Wozabal, 2010). That work made a review about single-state portfolio selection and multiple-period portfolio selection. Stochastic optimization considers parameters in the model to be probability distributions instead of deterministic ones. Monte-Carlo approaches and clustering methods are used to estimate these distributions. The conclusion was evolutionary algorithms allow considering a wide variety of risk measures and restrictions.

Some references (Adebiyi & Ayo, 2015) have proposed modifications to the differential evolution algorithm for portfolio problems. That reference presented the generalized differential evolution algorithm 3 (GDE3) to solve a mean-variance portfolio selection problem with the following restrictions: Bounded portfolio weights, cardinality, minimum transaction lots, and expert opinion. Expert opinion is a weight in the range $e_i = [0, 1]$ which describes the likelihood the asset to attain its expected return. Assets with $e_i < 0.5$ are not included in the portfolio. This value is randomly initialized for this study. The model includes the investor's desired return. GDE3 generates both feasible and unfeasible offspring. Feasible offspring are always preferred. On the other hand, their parents should dominate the offspring to be selected. Crowding is used to determine which individuals are located at less populated areas of solution space. Individuals are sorted according with this measure. Non-dominated and feasible individuals are saved for the next generation. The proposed method was tested with data from the Hong-Kong Hang Seng index and the German DAX-100. The method was tested with different desired return values and portfolio sizes. Mean variance of solutions, worst variance, standard deviation of solutions variance and mean execution time were the proposed performance measures. The method was compared against genetic algorithms, simulated annealing, taboo search (TS) and particle-swarm optimization (PSO). The proposed method outperformed the benchmarks in the experiments.

Ranković, Drenovak, Stojanović, Kalinić, and Arsovski (2014) proposed the solution of a Value-at-Risk (VaR) portfolio selection problem using genetic algorithms. That references stated the Pareto front of optimal portfolios is not restricted to be connected and convex when value-at-risk is considered. This work studied the proposed problem using a single-objective genetic algorithm and the multi-objective algorithm SPEA2. This work proposed the use of portfolio weights computed from the number of shares instead from the asset value. This modification avoids dynamic properties of markets to be mixed with static portfolio optimization. The single objective genetic algorithm combines both objectives using the parameter λ and its complement $1 - \lambda$, where $\lambda = [0, 1]$. $\alpha = 0.05$ to compute value-at-risk in the experiments. 10 exchange-trade funds (ETFs) were considered. Data was taken from February 2008 to December 2010. An equally-weighted buy-and-hold portfolio was used for comparison. Different values of λ were used to build the efficient frontier using the single-objective genetic algorithm. Two methods were used to obtain the efficient frontier: The first one used a fixed set of increasing λ values. The second compute the necessary λ value to attain a particular return level. This latter method obtained the best results. These results were also better than the ones obtained using SPEA2. The authors remarked solutions obtained using single-objective genetic algorithm were not necessarily Pareto optimal because value-at-risk is not a coherent risk measure. SPEA2 has the advantage to compute all the solutions simultaneously. All three methods outperformed the benchmark.

Trading (T)

Trading is the practice of finding profitable investment strategies. Forecasting is related to trading because an estimation of the future is usually required to make correct decisions. Trading is concerned about what to do with the forecast to make profit. For example, risk (i.e. uncertainty) can be considered to minimize loss probability along with maximizing expected return.

M. Lim and Coggins (2005) used genetic algorithms to find trading rules. The fitness function was based on the volume-weighted averaged price (VWAP) measure. The approach outperformed buy-and-hold, which is the strategy to keep a security until the end of time horizon. It was concluded volume information is useful for trading. A reinforcement learning approach was suggested in the future work section.

Some references (Hirabayashi, Aranha, & Iba, 2009) proposed to find the best trading time instead of forecasting. The estimation is based on technical analysis indicators like the relative strength index (RSI), moving averages, and percent difference from moving averages. It was applied to FOREX data for USDs, Euros, and Japanese yens. Positive returns were reported in stationary statistics time windows. A multi-objective approach was suggested in the future work section to include risk in the optimization.

Matsui and Sato (2009), proposed a solution based on genetic algorithms using binary and integer representation of rules. Technical indicators like moving averages, exponential moving averages, and Bollinger bands were used to conform these trading rules. In the presented experiments, the integer representation obtained higher profits at less computational cost. A later contribution (Matsui & Sato, 2010) extended this work suggesting the use of neighborhood evaluation in genetic algorithms for trading applications. The idea is to reduce over fitting by considering the average of neighbors and individual fitness. The experiments showed the method presented less over-fitting than the former approach, although the computational cost increased proportionally to neighborhood size. Further investigation of this approach was recommended in the future work section.

Lipinski (2012) described a similar approach for trading using technical indicators. Genetic algorithms and simulated annealing were proposed to solve the problem. Both techniques were improved with local search operators. A parallel processor architecture is described and tested in this work. Some references have combined the concept of portfolio optimization with trading to create investment methods. This is possible because a portfolio change implies a trading action (J.-S. Chen, Hou, Wu, & Chang-Chien, 2009). That work defined portfolio optimization as a combinatory optimization problem. A new combination genetic algorithm was suggested to solve the problem. Custom operators for combinatory problems were proposed.

Some references have studied the efficiency of traditional approaches (e.g. equally weighted portfolios) for trading (Sarijaloo & Moradbakloo, 2014). In that work, genetic programming was proposed to solve a mean-variance portfolio problem with maximum and minimum limits in their composition. The best 50 securities from the Tehran stock market index $(2006 - 2009)$ were chosen for this study. Optimal portfolios were chosen yearly and compared against equal-weight portfolios and random search. The approach showed better performance for the time range used in the experiments.

Yaman, Lucci, and Gertner (2014) used evolutionary programming (EP) to generate trading agents with different investment strategies. Agents were modeled using echo-state network models (ESNMs). They are a type of neural network with three layers: Input layer, hidden layers, and output layer. Feedback is connected from the output layer to the hidden layer and from the hidden layer to itself . All connection but the ones going to the output layer are randomly initialized and fixed from the beginning. Evolutionary programming is used to optimize the free connections. The individuals have two parts: The objective vector and the variance vector. Only the first one is evaluated, but both of them evolve with time. Mutation is the only operator used. This work used an EP($\mu + \mu$) type in the experiments. This means μ new individuals are generated from μ original individuals. The half of the population with the lowest fitness values is discarded. Resulting agents have different behavior depending on their initialization. Exchange rate from different currencies and technical indicator were used as inputs. The experiments concluded that having a set of different behavior agents is more profitable than one single type of agents.

Some works (Lohpetch & Corne, 2009) have stated other references who report profitable strategies were unable to repetitively outperform buy-and-hold. This work proposed a genetic programming algorithm to attain this goal. Technical analysis indicators are used for the rules. Some of the modifications to ensure solution quality were the following: The use of monthly data, a reduced function set, over-complexity penalization, and an objective function which penalized lower than buy-and-hold performance. The approach was successful in the presented experiments, although, the conclusion suggested conducting more tests with different time windows.

Y. Chen and Hirasawa (2010) proposed robust genetic network programming (R-GNP) to find trading strategies. A genetic relation algorithm (RGA) was proposed to find optimal portfolio to be managed using the R-GMP model. This approach has the property to work with graphs instead of trees. On the other hand, relational genetic network programming encodes trading strategies using graphs. Judgment nodes, processing nodes and delays are part of this graph. Judgment nodes represent conditions, processing nodes represent trading actions and delays represent the technical indicator used. This work used the β parameter as risk measure. β has its origins in the capital allocation pricing model (CAPM). β is the fraction of market risk borne by the portfolio. The initial portfolio was build using a relational genetic algorithm (RGA). The paper concluded that further tests are necessary to evaluate the performance of relation genetic network programming.

Other works (Hochreiter, 2015) have also proposed a method based on genetic programming for trading applications. This reference used genetic programming and sentiment indicators to find trading rules. Sentiment indicators were extracted from a social media service to determine the expectations of the online trading community. The estimations are based on the number and contents of messages in the service about stock from the Dow Jones industrial average. If-then type rules were obtained with this method. Dow Jones data from 2010–2013 were used for training. The classical portfolio optimization model was used to build buy-and-hold portfolios. Equal weighted portfolios were also used for comparison. The proposed approaches outperformed both benchmarks in the experiments. The inclusion of transaction costs was left as future work.

Other references which proposed combinations of technical indicators and evolutionary algorithms were found in the literature (Radeerom, 2014). This reference is an example. The system generates a trading signal to indicate the best moments to buy or sell stock. The process consists of two phases: The first one is the selection of the most suitable stocks, the second is the trading of the selected stocks. The technical indicators used are the following: Relative strength index (RSI) and moving average convergence/divergence. The algorithm maximizes the last-day Sharpe's ratio. This risk-weighted measure allowed to treat the multi-objective problem as single objective. Stocks with negative Sharpe's ratio or negative shareholder's equity value are eliminated. Trading rules based on the mentioned indicators and their combinations are optimized with training data. The approach was tested with the Thai-100 stock index. The results showed the proposed method outperformed buy-and-holds.

A.-P. Chen and Chang (2005) proposed the use of the XCS for trading. The rules were encoded to process sentiment indicators. Sentiment indicators are variables which measure general expectation of investors about market trend. For example, a bull market means investors expect a rise in prices, while bear markets indicates price drops. The volatility index, put-call ratio, and trading index were used in this work. The XCS found rules to decide when to sell or by futures. It was compared against buy-and-hold (the future was kept until last day), a trend-based strategy and a mean-reversion strategy. This last strategy made transactions when sentiment indicators reached a threshold. The XCS showed better performance than the other strategies.

H. Huang, Pasquier, and Quek (2009) proposed a hybrid system which combines a hierarchical co-evolutionary fuzzy System (HiFECS) and a hierarchical co-evolutionary genetic algorithm (HCGA) to forecast stock prices. A prudent strategy based on the price percentage oscillator (PPO) is run using these forecasts. The experiments showed HiFECS outperformed buy-and-hold and other predictive models like evolving neural networks (EFuNNs), dynamic neural-fuzzy inference systems (DENFIS), and rough set-based pseudo outer-product fuzzy neural networks (RSPOP).

Estimation of distribution algorithms have also been considered in some references (Lipinski, 2007). This work made a comparative study about the extended-compact genetic algorithm (ECGA) and BOA. Some modifications were necessary to perform online trading. In the experiments, both algorithms attained better returns than static strategies like buy-and-hold. Although, ECGA proved to be time-consuming and not suitable for an online application. BOA was faster, but with lower returns.

Matsumura and Kakinoki (2014) proposed a multi-objective genetic programming approach for portfolio-based trading. The algorithm first determines the 10 most suitable stocks to be traded. A multi-objective genetic algorithm is proposed for this end. The algorithm is a regular genetic algorithm where the fitness value is the number of individuals which dominate the individual to be evaluated. The average return along the time horizon and covariance are used to determine domination of individuals. Elitism and an end-cutting operator ensures solutions to be non-dominated and well spread. This approach is used in a second step for portfolio selection. Investment ratio is the fitness value to be maximized. This step uses a real-numbered representation of individuals. A third step uses genetic programming to optimize strategy trading trees. In this case, an optimal initial portfolio is selected and buy/sell operations rules are determined based on technical indicators. Experiments used data from the Nikkei-225 index. The results were compared against the performance of buy-and-holds. The performance of the proposed technique varied with the market current scenario.

Hu, Feng, Zhang, Ngai, and Liu (2015) proposed a XCS was to find trading rules based on trend following strategies (TF). Technical indicators were computed from history data to input the system. Moving averages and volume moving averages were used for this end. Their state is binary encoded for each stock. The XCS estimates short-term trading only, while long-term trend is estimated using moving averages only. Both signals are combined to generate the final trading signal. Buying occurs only when the XCS recommends it at bull market condition. Selling occurs when the XCS recommends a sell at bear market condition. No new position is opened while there is one already on course. Stop loss was implemented to avoid heavy loss. Data from the Shangai stock exchange from January 1, 2001 to July 31, 2013 were used for the tests. Performance was measured using the Sortino's ratio, which is a risk-weighted return measure where rise fluctuation is distinguished form fall fluctuation. Buy-and-holds, neural networks and decision tree models were implemented for comparison purposes. The XCS method attained higher return than the benchmarks. Besides, the generated rules were analyzed to reach some conclusion about the best behavior at different market conditions. The conclusion was that the generated rules were consistent with general financial knowledge.

Schmidbauer, Rösch, Sezer, and Tunalioğlu (2014) studied the effectiveness reduction suffered by trading rules-based systems when they are tested with out-of-sample data. The authors indicated this is a consequence of data-snooping bias. An a-priori robustness strategy is proposed to reduce this adverse effect. This approach used general programming, which encodes rules using a Backus-Naur form grammar. Fitness of individuals is computed from their profit and their hit-miss ratio. The approach requires opening, minimum, maximum, and closing prices to be input to the algorithm. A-priori robustness is ensured by generating simulated realizations of these time series. Time series realizations for closing are created using maximum entropy bootstrapping. Minimum and maximum signals are generated from ARMA models. The opening signal realizations are generated using kernel density estimation. The method was tested for FOREX trading of Euro/USD exchange with intra-day data from February to June 2011. A-priori robustness genetic programming was tested against regular genetic programming. The method was tested with different out-of-sample data, and different sets of set-up parameters. The results indicated the implementation of a-priori robustness and the time period chosen for the test were the main sources of variation for final profit. In-sample data robustness was curbed by the method, but it enhanced robustness of the out-of-sample tests. Profit seemed to be increased, but further work is needed to ensure the repeatably of the results. Fridays proved to be a day with higher uncertainty than others, the a-priori robustness method seemed to be more effective at that day.

Trading Execution (TX)

The trading execution problem is concerned with the methods used to fulfill trading orders efficiently. For example, a trading method could make the order to sell a stock which price has dropped dramatically, but the probability the actual order could be executed is low until stock price seems to stabilize. Another interesting case occurs when transaction volume is high enough to impact prices. Brokers should schedule partial orders to avoid adverse effects.

Almgren and Chriss (2001) suggested the problem of execution schedules. This approach considers investor wishes to liquidate a specific asset before some fixed time limit. Prices are affected by volatility, unbalance and market impact. The impact on the market depends on the volume of transactions. The authors used stochastic dynamic programming to find optimal execution strategies, and Monte-Carlo simulations to investigate their performance. Nevertheless, this work did not made use of evolutionary algorithms. Nevertheless, it was included in the review because of the novelty of the problem.

2.2.4 Evolutionary Algorithms and Financial Applications Review Conclusions

Results of the review are summarized in table 2.3, and figures 2.1 and 2.2. The surveys used to define the scope of this review were not considered. Table 2.3 presents the percentage of references which treated a particular problem and used a particular solution method. Open research areas can be identified as white spaces at the table. White spaces indicate the specific problem has not been solved with a specific solution method. Figure 2.1 summarizes total percentage of references per solution method. Figure 2.2 summarizes the total percentage of references per problem. Table 2.4, and figures 2.3 and 2.4 provide the same information about the references covered by other similar surveys. The references found in other surveys are called *past references* in this discussion. Care was taken to avoid counting repeated references. Most of the references of this review were published later in time than the scopes of other surveys. Some references which were published before the scope of other surveys were included for completion purposes.

Table 2.4 shows genetic algorithms are the solution approach with the widest application to solve different financial problems in the past. Table 2.3 shows this tendency has been continued. Figure 2.3 shows genetic algorithms references are about 45% of total. Genetic algorithms seemed to have become even more popular through time. Figure 2.1 shows genetic algorithms references augmented to about 60% of the total. The rise of interest in genetic algorithms is probably due their popularity has spread outside of the evolutionary computing community. This means people from other fields has adopted genetic algorithms to solve their respective problems. Other Darwinian approaches are yet in process to reach mainstream popularity outside the field. Genetic algorithms applications have also changed with time. MKS and BKR were the most popular applications, according with past references, but new references indicate that FC, T, and PSP are now more popular than MKS. This could be explained probably MKS was the first link between the economics community and the evolutionary computing community. Further research is needed to confirm it.

Multi-objective algorithms references and genetic programming references were at second and third place, respectively. Nevertheless, while the number of genetic programming references seems to remain unchanged (about 20%), multi-objective references percentage has dropped significantly. Although, the drop of percentage of references does not mean the community has lost interest in them. This drop seems a relative effect to the rise of interest in genetic algorithms. On the other hand, The number of problems approached with genetic programming and multi-objective algorithms seemed to have decreased with time. FC and T problems seemed to have substituted MKS for the genetic programming case. This could be explained by the reported advantages of its ability to build explainable rules. Genetic programming has been applied to problems where new knowledge about markets and investors is desired. MKS has benefited from this properties, but the idea to have several agents working simultaneously appears to be costly. Island genetic algorithms allow a similar effect with a single population. Islands could be implemented in genetic programming as well.

Learning classifier systems are the fourth approach with more references. Table 2.4 showed they have been used in the past for ABN and MKS. New references show the interest has changed to FC and T applications. The references only mentioned LCS and XCS types. Other variations of learning classifier systems have yet to be applied to financial problems. Nevertheless, figure 2.1 shows the interest for learning classifier systems in financial applications have augmented with time.

Figure 2.4 shows the percentage of total references per problem. MKS, PSP, and BKR were the three most addressed problems in past references. Newer references indicate the interest has changed to FC and T applications. One hypothesis about this change is, probably, PSP applications were re-defined to be trading problems in new references. Changes in portfolio are equivalent to trading decisions. The difference between PSP and T is the factor of risk, which is explicitly stated in PSP. Trading surely benefits from a return/risk model because it allows to include uncertainty into the optimization process. FC seemed to have attained higher levels of attention because better estimations of the future has direct impact in profit. Therefore, trading applications try to manage risk in the most profitable manner while FC applications try to reduce risk. In other words, T and FC interest raised together because they are closely related to each other.

Figures 2.1 and 2.3 show new references have payed more attention to co-evolutionary approaches and estimation of distribution algorithms. There is the possibility some of the referenced works has made use of co-evolution without explicitly report it. Co-evolution is more a concept than a concrete technique. This means there could be co-evolutionary approaches using genetic algorithms, genetic programming, or any other technique. The co-evolution term was found in past references, but these cases did not associated it to a evolutionary approach. Therefore, it could not be counted as such. Other cases did not clearly stated their evolutionary computing approach was co-evolutionary.

On the other hand, estimation of distribution algorithms are a relatively new approach. It seems natural they are yet to find application on different fields. Besides, their references are specially concerned with deception, linkage and hierarchical problems. A methodology to characterize real-world problems using these concepts seems an open question. Nevertheless, both co-evolution and estimation of distribution algorithms seem solution methods with high potential to be exploited in financial applications.

Figures 2.2 and 2.4 show there are problems which have received relatively few attention in the literature. TX, PRC, ITR, and CM are the problems with less references. Most of these problems seemed to be neglected also in past references. Darwinian approaches are a promising option to find solution to them.

Finally, tables 2.3 and 2.4 indicate financial applications of Darwinian approaches is yet open to further research. Both tables show many combinations of problems and approaches where no references were found. This occurs in both tables. Moreover, references in both tables are concentrated at few problem-approach pairs. This indicate even the problems already addressed are still open to further investigation.

Table 2.3: Summary of Open Problems. Percent of References per Problem and Approach.

Problem	GA	GP	LCS	MO	CV	EDA
ABN	2.778					
ARB	1.389	4.167				
BKR	1.389			1.389		
CS	6.944					
CM	1.389					
CP						
FA	2.778					
FC	20.833	6.944	2.778			1.389
ITR	2.778					
MKS	1.389				2.778	
PRC	1.389					
PSP	6.944	1.389		4.167		
T	8.333	8.333	1.389	2.778	2.778	1.389
TX						

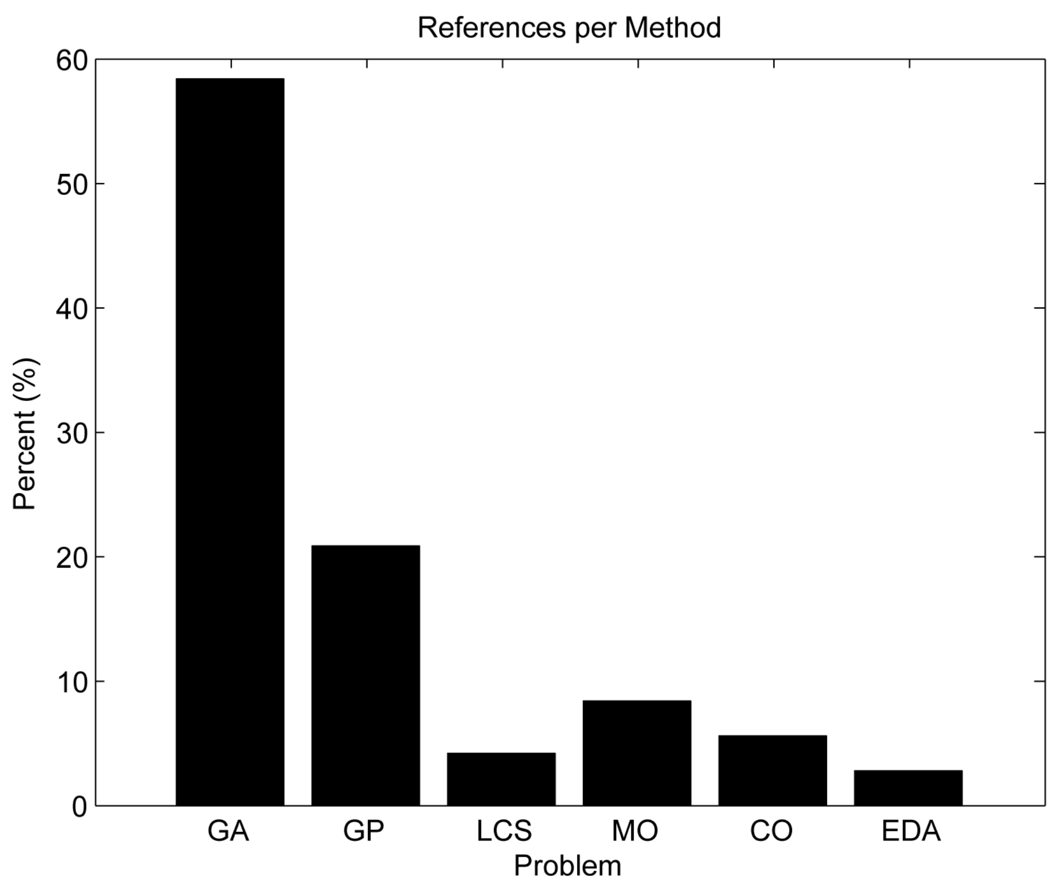

Figure 2.1: Summary of References per Solution Approach.

Table 2.4: Summary of Other Surveys. Percent of References per Problem and Approach.

Problem	GA	GP	LCS	MO	CV	EDA
ABN	3.846	0.769	0.769			
ARB	0.769					
BKR	16.154	3.077				
CS	2.308	1.538				
CM						
CP				2.308		
FA	0.769			2.308		
FC	3.077			2.308		
ITR						
MKS	15.385	10.769	0.769	4.615		
PRC						
PSP	0.769			18.462		
T	3.077	2.308		3.846		
TX						

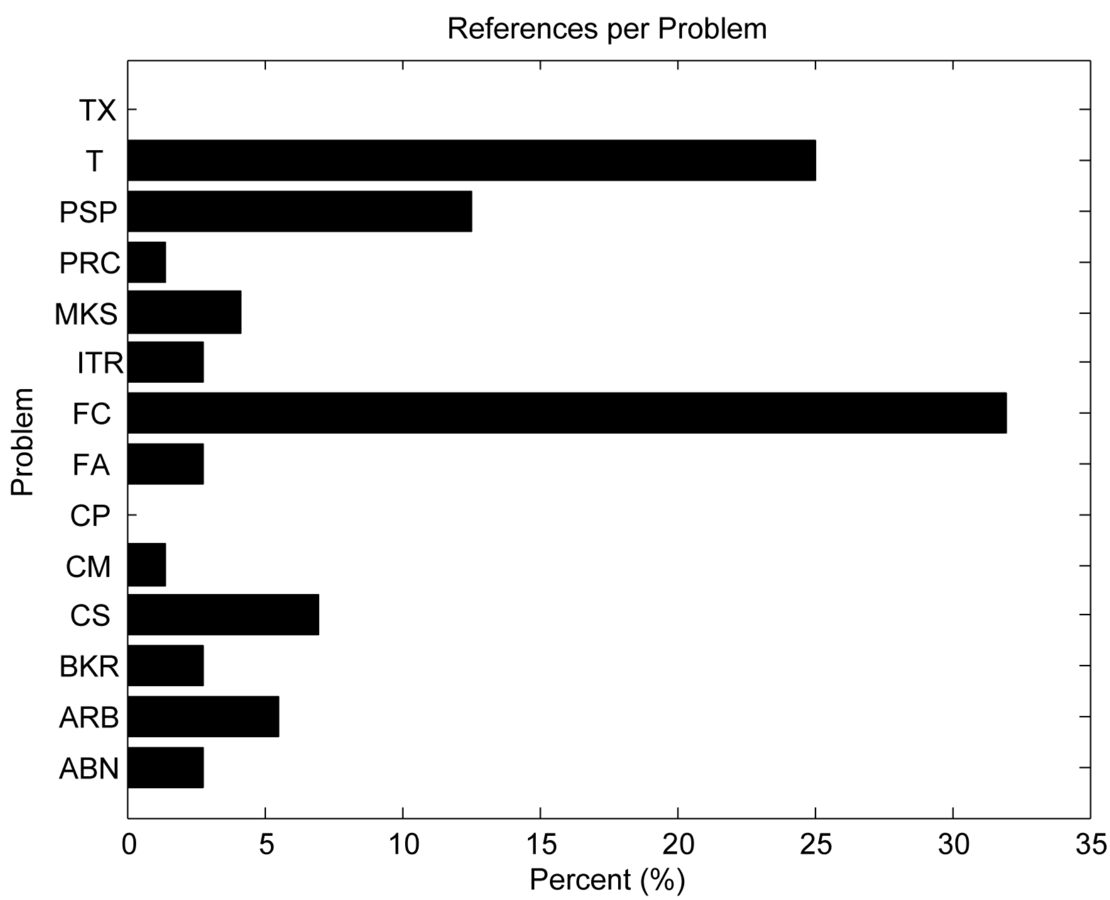

Figure 2.2: Summary of References per Problem

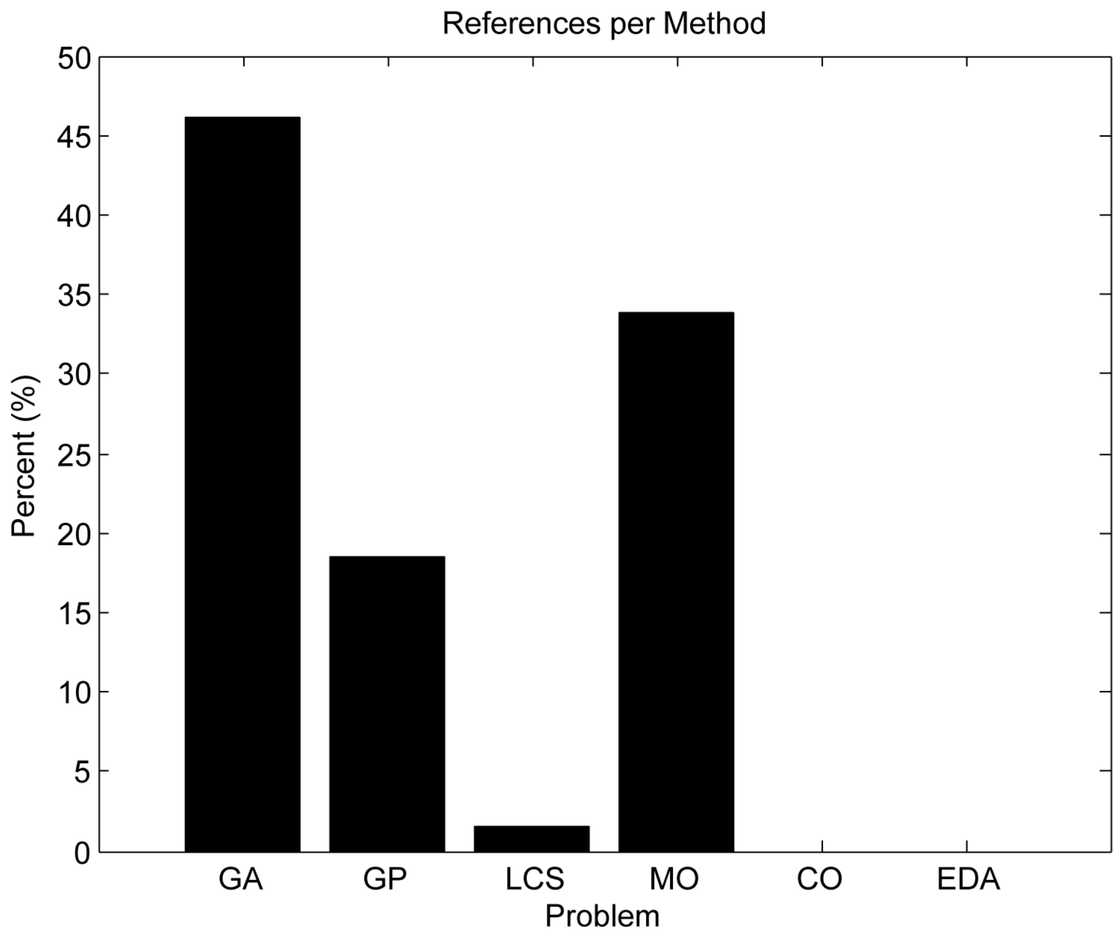

Figure 2.3: Summary of Other Surveys per Solution Approach.

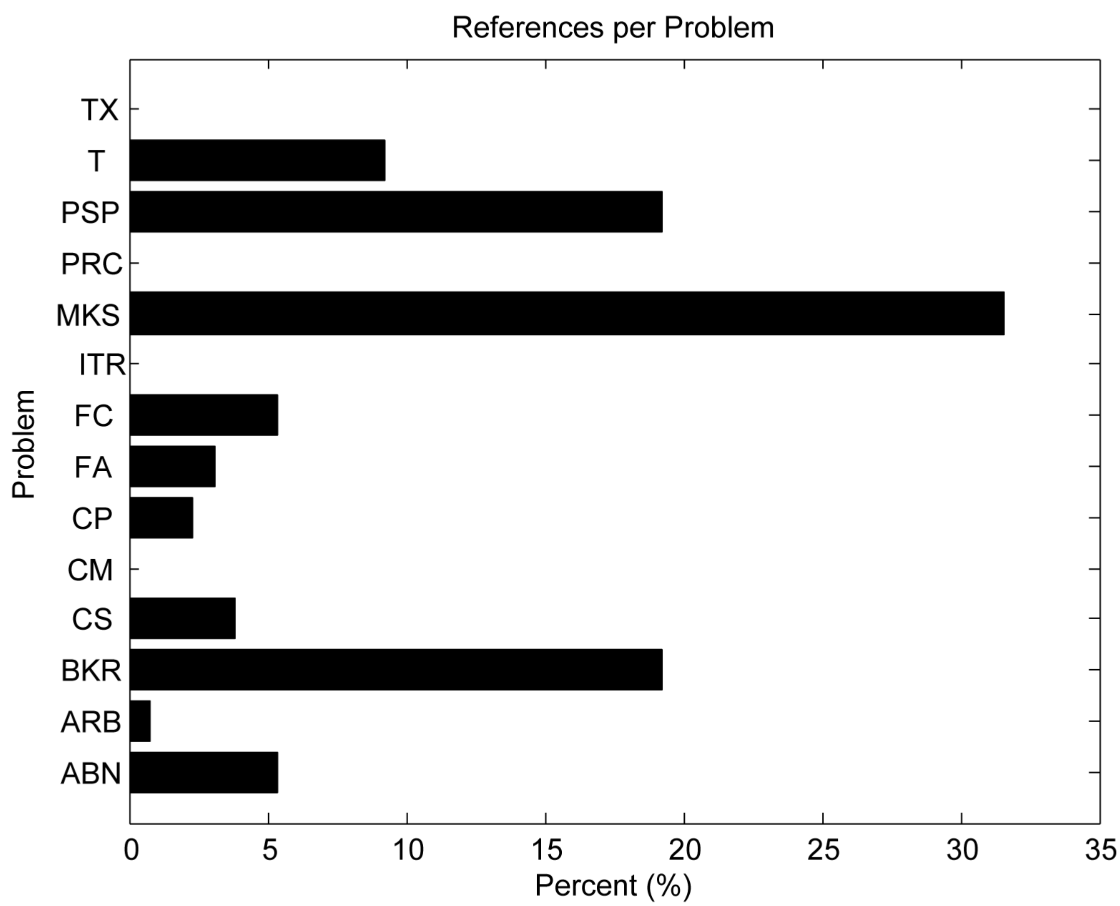

Figure 2.4: Summary of Other Surveys per Problem

2.3 Conclusions

This chapter presented the State-of-the-Art of the problem. It was divided in two parts. The first one focused on the solution of multi-period portfolio decision problems, and the second one was devoted to the application of evolutionary algorithms to finance.

The multi-period portfolio theory was proposed in the references to include transaction costs and other dynamic restrictions to portfolio optimization. The problem is relevant when state-dependencies are part of its definition. Most of the references are focused on finding closed-form solutions to the problem, which has proved to be a difficult task. Also, some reference has proposed numerical methods could be useful to find solutions in real-world applications. Extending the last idea, this work proposes using evolutionary algorithms to solve the problem applying Monte-Carlo methods. The evolutionary approach has the advantage to include any kind of restriction with less difficulty than purely mathematical approaches.

The second part was a review about evolutionary algorithms and financial problems. The review identified the coverage of other similar works and limited its scope to uncovered references. This approach allowed an analysis about the change of the problems and solution approaches along time. The review concluded the interest about problems and solutions approaches have changed with time, and genetic algorithms are the most popular approach. Also, it identified some research opportunities which deserve further attention.

The conclusion was multi-objective optimization has been widely used to solve portfolio selection problems, but multi-period portfolio selection have received limited attention. Therefore, this approach is a good suggestion to implement the solution model proposed in this work.

Chapter 3

Structure-based Evolutionary Algorithm

The hypothesis of this work states evolutionary algorithms are well suited to devise an investment method which considers dynamic restrictions, the investor's preferences, and data innovations to make decisions. The solution model considered a multi-period framework for the portfolio selection problem because it allows the inclusion of transaction costs, inflation, and other state-dependent factors into the optimization. Last chapter concluded multi-objective evolutionary algorithms (MOEAs) are a natural choice to solve this type of problems.

Multi-objective evolutionary algorithms have been mainly used to solve static portfolio selection problems with different constraints combinations in the literature. The multi-objective approach is preferred because it is able of finding several Pareto-optimal solutions in a single run. Several multi-objective algorithms have been proposed in the literature, although, this work introduces a new algorithm to implement the investment decision method.

The first part of the chapter makes a brief introduction to multi-objective optimization and evolutionary algorithms. The second part describes the new algorithm. The last section shows some tests about its performance. The explanation assumes the algorithm is applied to classical multi- objective optimization. Stochastic optimization will be developed in later chapters.

3.1 Multi-Objective Optimization

Multi-objective optimization deals with the problem of finding optimal solutions for many objective function simultaneously. The objectives can be contradictory. For example, the design of engines with maximum power and minimum fuel consumption imposes a conflict because fuel consumption increases with the engine's power.

In general, the specific problem addressed by evolutionary multi-objective algorithms is the following (Ponsich et al., 2013):

$$\left.\begin{array}{lll} \min & \mathbf{F}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})], & \\ \text{subject to} & g_j(\mathbf{x}) \leq 0, & j = 1, 2, \ldots, J; \\ & h_k(\mathbf{x}) = 0, & j = 1, 2, \ldots, K; \end{array}\right\} \tag{3.1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is the vector of decision variables, $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, \ldots, m$ are the objective functions, and $g_j, h_k : \mathbb{R}^n \to \mathbb{R}, j = 1, \ldots, J, k = 1, \ldots, K$ are the problem constrains.

Moreover, the optimality concept assumed by these algorithms is the following (Ponsich et al., 2013):

Definition 1 *Given two vectors* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{F}(\boldsymbol{x}) \leq \boldsymbol{F}(\boldsymbol{y})$ *if* $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ *for* $i = 1, \ldots, m$, *and* $\boldsymbol{x}$ **`dominates`** $\boldsymbol{y}$ *(denoted by* $\boldsymbol{F}(\boldsymbol{x}) \prec \boldsymbol{F}(\boldsymbol{y})$ *) if* $\boldsymbol{F}(\boldsymbol{x}) \leq \boldsymbol{F}(\boldsymbol{y})$ *and* $\boldsymbol{F}(\boldsymbol{x}) \neq \boldsymbol{F}(\boldsymbol{y})$.

Definition 2 *A vector of decision variables* $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$ *is* **`non-dominated`** *with respect to* $\mathcal{X}$*, if there does not exist another* $\boldsymbol{x} \in \mathcal{X}$ *such that* $\boldsymbol{F}(\boldsymbol{x}') \prec \boldsymbol{F}(\boldsymbol{x})$.

Definition 3 *A vector of decision variables* $\boldsymbol{x}^* \in \mathcal{F} \subset \mathbb{R}^n$ *(where* $\mathcal{F}$ *is the feasible region) is* **`Pareto Optimal`** *if it is non-dominated with respect to* $\mathcal{F}$.

Definition 4 *The* **`Pareto Optimal Set`** $\mathcal{P}^*$ *is defined by:*

$$\mathcal{P}^* = \{\boldsymbol{F}(\boldsymbol{x}) \in \mathcal{F} | \, \boldsymbol{x} \text{ is Pareto optimal}\}.$$

Definition 5 *The* **`Pareto Front`** $\mathcal{PF}^*$ *is defined by:*

$$\mathcal{PF}^* = \{\boldsymbol{F}(\boldsymbol{x}) \in \mathbb{R}^m | \, \boldsymbol{x} \in \mathcal{P}^*\}.$$

3.1.1 Classical Approaches

Some classical techniques to deal with multi-objective problems are the following (Deb, 2001):

- Weighted sum methods.
- ϵ-constraint methods
- Weighted metric methods
- Benson's Method
- Goal programming methods

The first method converts a multi-objective problem into a single-objective problem. A vector of weights $\mathbf{w}$ is used to combine the objectives as a summation of the form:

$$\left.\begin{array}{lcc} \text{min} & \sum_{m=1}^{M} w_m f_m(\mathbf{x}), & m = 1, 2, \ldots, M; \\ \text{subject to} & g_j(\mathbf{x}) \leq 0, & j = 1, 2, \ldots, J; \\ & h_k(\mathbf{x}) = 0, & k = 1, 2, \ldots, K; \\ & x_i^L \leq x_i \leq x_i^U, & i = 1, 2, \ldots, n; \end{array}\right\} \tag{3.2}$$

Where $f_m(\mathbf{x})$ is the m-th objective function from $\mathbf{F}(\mathbf{x})$ to be optimized. $[L, U]$ are the limits imposed to the value of $\mathbf{x}$. In equation 3.2, w_m is the m-th weight from the vector $\mathbf{w}$, which is used to combine the objectives. One single solution is found for a given $\mathbf{w}$.

The main difficulty of this approach is determining the weight values. They are chosen considering scaling factors and the importance of objectives. Nevertheless, this method suffers the inability of finding certain solutions for non-convex problems (Deb, 2001).

The ϵ-constraint method was proposed to alleviate these limitations. This method redefines the problem imposing a limit ϵ to all the objectives but one. The problem definition now includes this restriction:

$$\left.\begin{array}{lcr} \text{min} & f_u(\mathbf{x}) & \\ \text{subject to} & [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_{u-1}(\mathbf{x}), f_{u+1}(\mathbf{x}), f_M(\mathbf{x}),] \leq \epsilon & \\ & g_j(\mathbf{x}) \leq 0, & j = 1, 2, \ldots, J; \\ & h_k(\mathbf{x}) = 0, & k = 1, 2, \ldots, K; \\ & x_i^L \leq x_i \leq x_i^U, & i = 1, 2, \ldots, n; \end{array}\right\} \quad (3.3)$$

In equation 3.3, ϵ is not restricted to be a small quantity. The modified problem can be solved using mathematical methods (e.g. Lagrange multipliers). The method simplifies the problem because the search space is reduced by the new limits.

Weighted metric methods use metrics instead of sums to combine the individual objectives. An ideal reference point $\mathbf{z}^*$ is determined to compute the metrics. The new problem seeks to minimize the distances. The problem is now defined in the following manner:

$$\left.\begin{array}{lcr} \text{min} & \mathcal{L}_p(\mathbf{x}) = \left(\sum_{m=1}^{M} w_m \left|f_m(\mathbf{x} - \mathbf{z}_m^*)\right|^p\right)^{1/p} & \\ \text{subject to} & g_j(\mathbf{x}) \leq 0, & j = 1, 2, \ldots, J; \\ & h_k(\mathbf{x}) = 0, & k = 1, 2, \ldots, K; \\ & x_i^L \leq x_i \leq x_i^U, & i = 1, 2, \ldots, n; \end{array}\right\} \quad (3.4)$$

Where $p \in [1, \infty]$. The value of p determines the type of norm used. The Tchebycheff's norm allows to find all the optimal solutions. Norm rotations or dynamic change of reference points are applied to find undiscovered solutions. Nevertheless, previous knowledge is needed to determine the best reference points. Single-objective optimization is needed to provide this information.

The Benson's method consists on choosing $\mathbf{z}^*$ randomly from the feasible region. The method maximizes the sum of the non-negative distances from the solution to the reference. The problem is defined by

$$\left.\begin{array}{lcr} \text{max} & B(\mathbf{x}) = \sum_{m=1}^{M} \max\left(0, \left(\mathbf{z}_m^* - f_m(\mathbf{x})\right)\right) & \\ \text{subject to} & F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})] \leq \mathbf{z}^* & \\ & g_j(\mathbf{x}) \leq 0, & j = 1, 2, \ldots, J; \\ & h_k(\mathbf{x}) = 0, & k = 1, 2, \ldots, K; \\ & x_i^L \leq x_i \leq x_i^U, & i = 1, 2, \ldots, n; \end{array}\right\} \quad (3.5)$$

This method alleviates scaling problems and can be applied to non-convex instances. Although, more constrains are imposed to the problem and the new objective functions are usually non-differentiable.

Goal programming methods define targets for each objective function. The method search for the solution which evaluations match the targets. The method minimizes the deviations from the targets even if the solution is unfeasible. Targets can be treated as constraints. The optimization problem is turned into a constraint satisfaction problem (CSP). The problem is defined by

$$\left.\begin{array}{lll} \min & \sum_{j=1}^{M}(\alpha_j\rho_j+\beta_j\eta_j) & \\ \text{subject to} & f_m(\mathbf{x})-\rho_j+\eta_j=t_j & m=1,2,\ldots,M \\ & \mathbf{x}\in\mathcal{S} & \\ & \rho_j,\eta_j\geq 0 & j=1,2,\ldots,n. \end{array}\right\} \tag{3.6}$$

Deviations ρ and η allow the method to handle both less-than-equal and more-than-equal restrictions. Different methods minimize deviations in their own way. For example, lexicographic goal programming assigns priorities to goals, solving restrictions sequentially according with their priority values. The method stops when a solutions satisfies all the restrictions. On the other hand, min-max goal programming minimizes the maximum deviation from each target.

3.1.2 Evolutionary Computation Approaches

Past chapters mentioned some multi-objective evolutionary algorithms proposed in literature. Their ability to optimize simultaneously many solutions is one of their advantages. Although, traditional genetic algorithms generally converge to one solution. Therefore, multi-objective algorithms include mechanisms to preserve different solutions and promote their spread along the Pareto front.

The literature reports several examples of multi-objective evolutionary algorithms. The first one reported is probably the vector evaluated genetic algorithm (VEGA) (Schaffer, 1985). VEGA was designed to maintain different solutions, according with each objective. The niched Pareto genetic algorithm was proposed by Horn, Nafpliotis, and Goldberg (1994). This algorithm used the idea of domination along with sharing to find different Pareto-optimal solutions. Sharing is applied to the non-dominated solutions, modifying their fitness values according with the distance among them. Individuals located at crowded areas are penalized. Sharing is defined by the following function:

$$\text{Sh}(d_{ij})=\begin{cases} 1-\left(\frac{d_{ij}}{\sigma_s}\right)^2 & \text{if } d_{ij}<\sigma_s \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

Where d_{ij} is the distance between the individuals. σ_s is called maximum niche distance. Individuals beyond this point are not considered for sharing calculations. This parameter should be adjusted by the user.

Fonseca and Fleming (1993) introduced the multi-objective genetic algorithm (MOGA). It uses a ranking system to evaluate individuals. Non-dominated individuals have a rank 1, while dominated individuals have a rank proportional to the density of the population. Sharing is used for diversity of Pareto solutions.

J. Knowles and Corne (1999) proposed the Pareto archived evolution strategy (PAES). This is an $(1+1)$ evolutionary strategy. This means the algorithm generates a single offspring per cycle and it comes from a single parent. The worst individual from the extended population is deleted. Besides, the algorithm relies on an external archive to keep the best non-dominated solutions. Special rules are needed to manage this archive effectively.

The Pareto envelope-based selection algorithm (PESA) is another example of this type of methods (Corne, Knowles, & Oates, 2000). It uses an external archive to save the optimal solutions, and the population is deleted each generation and generated from random members of the external archive or their offspring. Only non-dominated individuals are saved to the archive. Diversity is promoted using a grid. Individuals from low populated sections of the solution space are preferred for reproduction.

Zitzler et al. (2001) proposed the strong Pareto evolutionary algorithm (SPEA), and later a second version: SPEA-II. The fitness of individuals is computed from the fraction of population dominated by them (i.e. strength). It also implements an archive to save the non-dominated solutions. SPEA-II modified the definition of strength to consider both dominated and non-dominated individuals, prevented bounder solutions to be deleted from the archive, and provided a density estimation to promote diversity.

Deb et al. (2002) proposed two versions of non-dominated sorting genetic algorithm (NSGA). The fist version relied on sharing to spread solutions along the Pareto front. NSGA-II was created to appease the criticism about the computational complexity and the lack of elitism of the first version. NSGA-II uses non-domination sorting to compute fitness and crowding to promote the diversity among non-dominated solutions. Elitism allows the best solutions to be always included into the current population.

Laumanns and Ocenasek (2002) proposed a combination of the NSGA-II with the Bayesian optimization algorithm (BOA) to obtain MO-BOA. MO-BOA uses a Bayesian network to generate new individuals in the same fashion BOA does. This algorithm has been tested against multi-objective versions of deceptive and linked problems.

Ponsich et al. (2013) mentioned other MOEAs besides the ones described: AbYSS, FastPGA, IBEA, and MOCeLL are some examples. AbYSS uses a $(1+1)$-Evolutionary Strategy to search solutions and includes an external archive. Crowding is used for density estimation in a fashion similar to PESA. FastPGA proposes an archive of dynamic size to avoid rejection of non-dominated solutions. IBEA (Indicator Based Evolutionary Algorithm) uses arbitrary performance measures which are defined by the user. MOCeLL uses an external archive and random individuals are deleted from the population and substituted by random archived individuals. A neighborhood is defined to determine which individuals can be paired for reproduction. Crowding is used to determine how to delete individuals when the archive is full.

3.2 Proposed Method

The review showed multi-objective evolutionary algorithms share some common features: Mechanisms to preserve multiple solutions and their spread along the non-dominated front, and methods to store the optimal solutions. Some algorithms define an external archive to store solutions. Others, like NSGA-II, do it implicitly through elitism. The proposed algorithm, called structure-based multi-objective evolutionary algorithm (Sb-MOEA), includes these features in the following manner:

- Solutions are stored using a non-generational population scheme.

- Multiple solutions are preserved using a fitness measured based on Pareto structure. This measure is called *Area measure* A_{ms} in this work.

- Diversity is promoted using an alternative crowding measure (C_{ms}) proposed in the literature (Köppen & Yoshida, 2007). Crowding becomes useful when most of individuals are non-dominated. This happens at the last stages of execution.

Algorithm 3.2 shows the pseudocode for Sb-MOEA. Details about the implementation of these features are presented in the following subsections.

Algorithm 1 Sb-MOEA Pseudocode

```
function SB-MOEA(N,M,fobj,Ncyles)
    pop = INIT_POPULATION(N,M)
    for i = 1:Ncycles do
        pop = EVALUATE(pop,fobj)
        ams = AREA_MEASURE(pop)
        cms = CROWDING(pop)
        cnms = STOCHASTICFIT(pop,cms)
        parents = SELECTION(pop,tokill,ams,cnms)
        offspring =CROSSOVER(parents)
        offspring =MUTATION(offspring)
        pop = NG-CYCLE(pop,offspring)
    end for
    return pop
end function
```

3.2.1 Non-generational Genetic Algorithm

Non-generational schemes can be found early in the literature. Whitley (1989) proposed Genitor, where the weakest individual is deleted from the population and substituted by a new one. Linear ranking selection was used to determine the parents of the new individual. Goldberg and Deb (1991) determined non-generational schemes have higher selective pressure, which could be beneficial in some cases. Non-generational multi-objective genetic algorithms were also reported in the literature (Valenzuela-Rendón & Uresti-Charre, 1997).

Non-generational schemes have the advantage to preserve individuals. Therefore, non-generational populations can be used to save non-dominated solutions in the same way archives do. Besides, this approach simplifies storage management, which is a problem in archived-based methods. Elitism is implicit in non-generational populations, which is considered beneficial to multi-objective evolutionary optimization (Deb et al., 2002).

Sb-MOEA computes fitness from the sum of A_{ms} and C_{ms}. The two weakest individuals are deleted from the population. Two parents are chosen using roulette wheel, and their children are included in the population. A pair is chosen to allow wider exploration of search space per cycle.

In algorithm 3.2, Evaluate( ) is used to evaluate the objective function. Functions Selection( ), Crossover( ), Mutation( ), and NG-Cycle( ) implement the non-generational genetic algorithm.

3.2.2 Area measure ($A_{\mathbf{ms}}$)

The non-domination sorting method, used in the NSGA-II (Deb et al., 2002), gives an integer rank to each individual based on the number on the domination relationships between them. As a consequence, a large number of individuals in the population have the same fitness value, making impossible to determine which one is better from them. On the contrary, the proposed metric intends give a continuous fitness value to each individual based on its relative position in the solution space and the positions of the non-dominated individuals. This method intends to avoid ties between individuals and provide a better description of their closeness to the non-dominated front.

The approach is illustrated in figure 3.1. The solid line represents the non-dominated front, and the points are solutions. The term *non-dominated front* is used to indicate this one is not necessarily the true Pareto front. Non-dominated solutions are labeled by roman numbers, and dominated solutions are labeled using capital letters. The shaded areas denote the region dominated by a particular solution. Darker tones indicate that particular area is dominated by more individuals than clear toned areas.

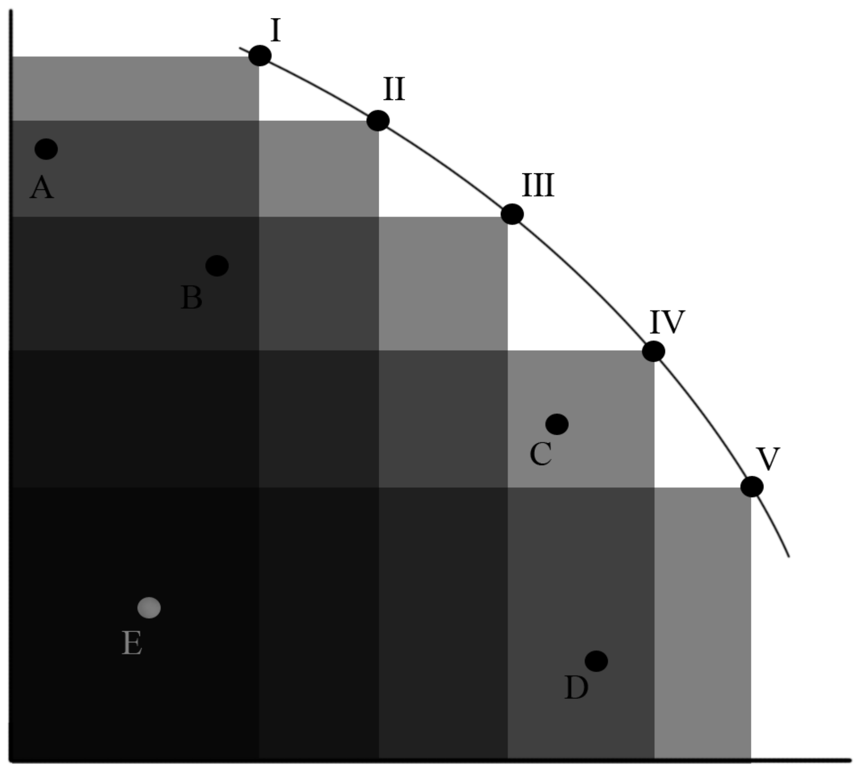

Figure 3.1: Structure of Pareto Front and Dominated Individuals.

In figure 3.1, we can observe the areas close to the non-dominated front have clearer shades than the ones far away from it. This occurs because areas covered by non-dominated individuals overlap more frequently with each other at far away areas from the non-dominated front. The information provided by the position of non-dominated individuals can be used to guide the optimization process. Individuals located at clear-shaded areas are favored because this indicates they are closer to the non-dominated front.

In this example, C is the best solution because it is only dominated by IV.A is dominated by I and II while D is dominated by IV and V. B is dominated by I, II, and III. E is dominated by all of them. A and D would have the same fitness value under this criterion.

Also, the spread of individuals is promoted by this method. Highly populated sections of the non-dominated front will overlap more frequently than sections with fewer individuals. For example, if II was not present, then both A and C would be dominated by only one solution. Besides both B and D would be dominated by two solutions. In this case, both A and B were benefited because they were covered by a less populated section of the non-dominated front. The method assumes the empty sections will be covered by the offspring of the favored individuals. Similar underlying assumptions are common in other multi-objective algorithms.

In the example, both A and D have the same fitness value. A method merely based on the number of dominating individuals could have a limited resolution. To overcome this problem, the area covered by the dominating individuals is computed instead to have a continuous fitness value. To make a difference for the cases with the same number of dominating individuals, the area covered by the evaluated individuals is also considered by the method. In this case, D covers a larger area of the solution space than A, then D would have a higher fitness value. Therefore, an expression for the proposed fitness measure is the following:

$$A_{\text{ms}}(J_k) = \begin{cases} \dfrac{\prod_{i=1}^{M} J_{ki}}{\prod_{j=1}^{\text{Nd}_k} \prod_{i=1}^{M} J_{ji}}, & \text{if } \text{Nd}_\text{k} > 0 \\ 1, & \text{if } \text{Nd}_\text{k} = 0. \end{cases} \tag{3.8}$$

In this work, the individuals from the non-dominated front which dominate a particular solution will be called *dominants*. The ratio of areas (or hyper-volumes) covered by the dominated individual and their dominants can be used to measure the degree of closeness to the Pareto front and population density of the individual. Equation 3.8 computes the product of the areas instead of the unified area, but it behaves in the same manner than the exact calculation. A non-dominated individual has $A_{\text{ms}} = 1$. Duplicated, non-dominated individuals receive an A_{ms} value inversely proportional to the number of existing copies in the population. This rule prevents the population converge to a single solution. Nevertheless, the A_{ms} from one of these individuals is computed using equation 3.8 to avoid the loss of valuable schemata.

In this method, suitable boundaries are needed to compute the areas (or hyper-volumes). In figure 3.1, the axis delimit these boundaries. The final implementation exempts the user from the difficulty to define suitable boundaries to compute A_{ms}. The idea is based on dynamic weighted metric methods, where the reference point changes accordingly with the current conditions of the problem. In this case, the user should provide the vector $\mathbf{z}_0$ where

$$\mathbf{z}_0 = [z_0(1), z_0(2), \ldots, z_0(M)], z_0(i) \in \{0, 1\}. \tag{3.9}$$

In this case, $\mathbf{z}_0$ is used to indicate the algorithm if objectives should be maximized or minimized. Maximization of objectives is denoted by 1, and minimization by 0. The algorithm computes the suitable magnitude vector $\mathbf{z}_{\text{mag}}$ in the following manner:

$$\begin{aligned} \mathbf{z}_{\text{mag}} &= [z_{\text{mag}}(1), z_{\text{mag}}(2), \ldots, z_{\text{mag}}(M)], \\ \mathbf{z}_{\text{mag}}(i) &= \max\left(\mathbf{J}_{\text{NM}}(j,i)\right), j = 1, 2, \ldots, N. \end{aligned} \tag{3.10}$$

In equation 3.10, z_{mag} contains the maximum value of each objective and $\mathbf{J}_{\text{NM}}$ is a matrix which contains the fitness values of all the objectives for each individual from the population. $\mathbf{J}_{\text{NM}}$ is dynamically transformed each cycle using $\mathbf{z}_0$ and $\mathbf{z}_{\text{mag}}$. The transformation allows the algorithm to perform maximization of all the objectives regardless of the original problem. Computations from different cycles are comparable because A_{ms} is normalized. $\mathbf{J}_{\text{NM}}$ is modified to be

$$\mathbf{J}'_{\text{NM}} = \left|\mathbf{J}_{\text{NM}} - 2\left(\mathbf{z}_{\text{mag}} \odot \mathbf{z}_0\right)\right|. \tag{3.11}$$

Where the $\odot$ operator denotes element-wise product of vectors or matrices. Equation 3.11 describes a coordinate transformation of the original fitness values. A new origin is defined from the product of z_{mag} and z_0. The new coordinates are translated to locate this point at the origin of the new coordinates plane. The absolute value of the coordinates is computed to flip the fitness values to the first quadrant of the new coordinates system. This transformations allows the problem to be handled as a maximization problem regardless of the original definition. The new axis are the boundaries required to compute A_{ms}.

3.2.3 Crowding Measure (C_{ms})

The area metric A_{ms} gives preference to uncrowded individuals which are close to the non-dominated front. Nevertheless, it offers no further guidance once most of the individuals become non-dominated. A crowding measure is useful for this end. For a non-dominated individual, the total fitness is the sum of both area measure and crowding measure. For dominated individuals, C_{ms} will be 0.

This work will use the term *crowding* to refer other population density metrics as well. For example, the called $-\epsilon$-domination (negative ϵ) method computes the smallest value which will turn the evaluated individual (which is non-dominated) into a dominated individual (Köppen & Yoshida, 2007). This value decreases with the distance from the evaluated individual to their neighbors, therefore, it can be used as a density metric. A definition of $-\epsilon$-domination is shown below (Xia, Zhuang, & Yu, 2014):

Definition 6 *if $f_i(\boldsymbol{x}) + \epsilon_i \leq f_i(\boldsymbol{y})\, \forall\, i = 1, 2, \ldots, m$ and $f_i(\boldsymbol{x}) + \epsilon_i < f_i(\boldsymbol{y})$ for at least one objective, we say* ***x* `negative epsilon-dominates` *y*** *(denoted by $\boldsymbol{x} \prec_{-\epsilon} \boldsymbol{y}$).*

Köppen and Yoshida (2007) proposed this metric to substitute crowding in the NSGA-II and improve its performance in many-objective problems. Their results showed $-\epsilon$-domination could contribute to find better distributed non-dominated fronts. An algorithm to compute $-\epsilon_i$ is the following:

Algorithm 2 Crowding Pseudocode

```
function CROWDING(J_NM, k, N, M)
    if N == 1 then
        eps = 1
    else
        eps = ∞ * ones(1, N)
        for i = 1 : N do
            for j = 1 : M do
                if i ≠ k then
                    eps(i)=MAX(eps(i),|J_NM(i, j) − J_NM(k, j)|)
                end if
            end for
        end for
    end if
    return MIN(eps)
end function
```

This algorithm computes ϵ_i as the minimum of the maximum differences from individual i to the rest of the population for each objective. Normalizing is used to avoid scaling problems. These calculations are computed in the Crowding( ) function.

3.2.4 An Example

The example is defined for maximization of all the objectives. This allows using the coordinates axis to define the areas covered by the individuals and makes them easier to visualize. This change can be done without loss of generality. In figure 3.2, the non-dominated individuals are denoted by roman numbers. The coordinates of each individual are the following: $A = [1, 4]$, I $= [2, 6]$, II $= [5, 5]$, and III $= [6, 3]$. A_{ms}(I) $= A_{ms}$(II) $= A_{ms}$(III) $= 1$ because they are the non-dominated individuals from the population. Equation 3.8 is applied to compute $A_{ms}(A)$. In this case, Area$(A) = 1 \times 4 = 4$, Area(I) $= 2 \times 6 = 12$, and Area(II) $= 5 \times 5 = 25$. Area(III) is not necessary because III $\not\succ A$. Therefore, $A_{ms}(A) = \frac{4}{12 \times 25} = \frac{1}{75}$.

On the other hand, $C_{ms}(A) = 0$ because A is a dominated individual. The difference between individuals is defined as $\Delta_{a-b} = |J_a - J_b|$. Therefore $\Delta_{\text{I-II}} = |[2, 6] - [5, 5]| = [3, 1]$, $\Delta_{\text{I-III}} = |[2, 6] - [6, 3]| = [4, 3]$, and $\Delta_{\text{II-III}} = |[5, 5] - [6, 3]| = [1, 2]$. In a similar fashion, $\Delta_{\text{I-A}} = [1, 2]$, $\Delta_{\text{II-A}} = [4, 1]$, and $\Delta_{\text{III-A}} = [5, 1]$. For individual I, the relevant differences are $\Delta_{\text{I-II}}$, $\Delta_{\text{I-III}}$, and $\Delta_{\text{I-A}}$. The maximum differences for each objective are computed and the result is $[4, 3]$. The minimum from the vector is $-\epsilon_{\text{I}} = \min[4, 3] = 3$. The values $-\epsilon_{\text{II}} = 2$ and $-\epsilon_{\text{II}} = 3$ are found in the same manner. These value are finally normalized based on the maximum $-\epsilon$ from the population, therefore, C_{ms}(I) $= C_{ms}$(III) $= 1$, and C_{ms}(II) $= \frac{2}{3}$.

3.2.5 Stochastic Fitness Mechanism

Density measures have proved to be useful to find well-spread solutions. These measures promote individuals to be more separated from each other. In this case, when new individuals

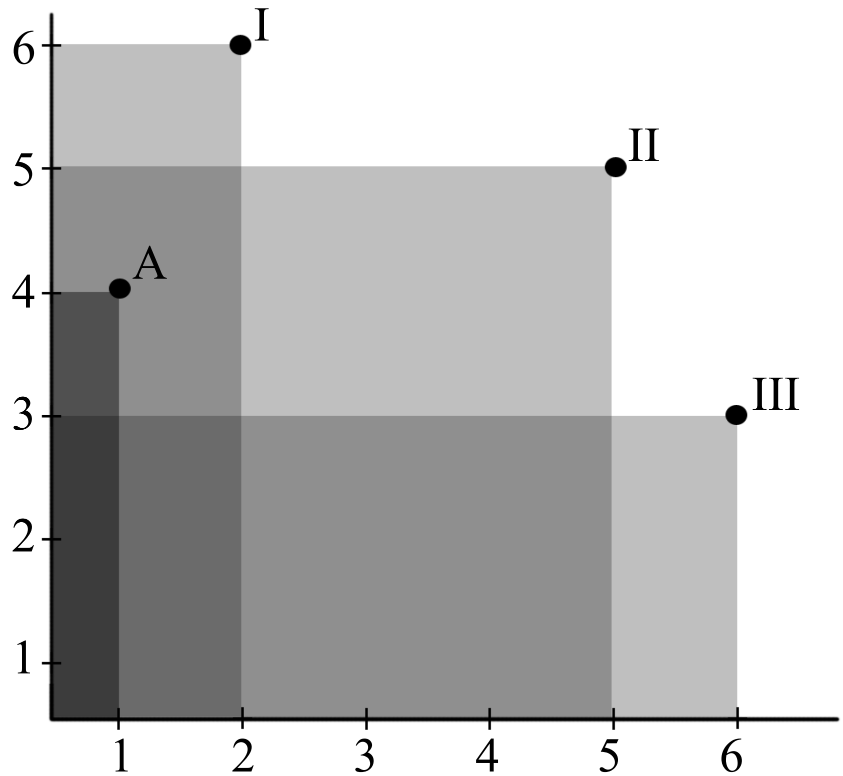

Figure 3.2: Example of the Application of Area Measure and Crowding Measure.

appears at a low-populated section of the solution space they will be favored with higher fitness values. As a consequence, these individuals will have higher chances to reproduce. If the parent is non-dominated and its offspring also appear at these low-populated areas, they will be eventually "fill the holes" at the non-dominated front and the spread will be improved.

Nevertheless, the assumption that the offspring will have similar evaluation than their parents cannot be assured, specially when high non-linear relationships are present in the search space. This situation limits the power of crowding measures because they cannot fully direct the search towards unexplored areas. Estimation of distribution algorithms (e.g. BOA) try to discover the nonlinear relationship between the chromosome's bits to overcome this difficulty (Pelikan et al., 2000a).

A simpler way is proposed to obtain similar effects. This method adds noise to the fitness value of non-dominated individuals to stimulate the algorithm to keep looking for better solutions. The magnitude of noise should be small to avoid the algorithm to become a random search. This procedure is inspired by the literature; some references have reported that a low level of noise helps genetic algorithms to escape local optima (Branke & Schmidt, 2003). Besides, genetic operators induce noise to fitness optimization process naturally. A uniform distribution $\mathcal{U}$ was preferred to avoid bias towards some particular group of individuals. δ controls the magnitude of induced noise. Noise is added to C_{ms}. The noisy crowding measure

$C_{\text{nms}}(k)$ is computed in the following way:

$$C_{\text{nms}}(k) = C_{\text{ms}}(k) + \mathcal{U}(0, \delta). \tag{3.12}$$

This quantity is normalized using the maximum value from the vector of population noisy crowding measures $\mathbf{C}_{\text{nms}}$:

$$C^*_{\text{nms}}(k) = \frac{C_{\text{nms}}(k)}{\max(\mathbf{C}_{\text{nms}})}. \tag{3.13}$$

The value of fitness f_k is computed from both A_{ms} and C^*_{nms}.

$$f_k = A_{\text{ms}}(k) + C^*_{\text{nms}}(k). \tag{3.14}$$

This calculation is applied to non-dominated individuals only. The rest of individuals have $C_{\text{ms}} = 0$. $A_{\text{ms}}(k)$ refers to the area measure of k-th individual of the population. Function StochasticFit( ) implements this feature.

3.3 Experiments

Sb-MOEA was tested against benchmarks proposed in the literature (Deb, 2001). These problems were used to test performance of NSGA-II, PAES, and SPEA in other references (Deb et al., 2000). Sb-MOEA was tested using some of these problems and compared against the results reported in the latter reference. The comparison was done using the Υ-metric and the Δ-metric proposed by Deb et al. (2000).

3.3.1 Test Problems

Deb et al. (2000) presented a series of problems to test the NSGA-II and other multi-objective algorithms. Although, not all them had a closed-form solution, turning any comparison cumbersome. Therefore, these problems were discarded to compare Sb-MOEA. A pair of test problems was selected for the experiments: The Scheaffer's problem (SCH) and Fonseca's (FON) problem. These problems include convex and non-convex instances. The SCH problem is defined in the following equation:

$$SCH(x) = \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \\ x \in [-10^3, 10^3]. \end{cases} \tag{3.15}$$

The solution to equation system 3.15 is $x \in [0, 2]$. This is a convex problem. On the other hand, the FON problem can be defined by the following equation system:

$$FON(x_1, x_2, x_3) = \begin{cases} f_1(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right) \\ f_2(x) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right) \\ x_1 \in [-4, 4] \\ x_2 \in [-4, 4] \\ x_3 \in [-4, 4]. \end{cases} \tag{3.16}$$

The solution for the equation system 3.16 is $x_1 = x_2 = x_3 \in [-1/\sqrt{3}, 1/\sqrt{3}]$. This is a non-convex problem.

3.3.2 Performance Metrics

The performance was measured with the Υ-metric and the Δ-metric. The Υ-metric measures the distance from the non-dominated front to the Pareto front, and the Δ-metric measures the spread of non-dominated solutions. The Υ-metric is the average of Euclidean distances from the non-dominated individuals to the closest optimal solution. Closed-form solutions allow to compute the Υ-metric exactly. An algorithm to compute the Υ-metric from the matrix of fitness evaluations $\mathbf{J}^*_{\mathrm{NM}}$ is shown below. $\mathbf{J}^*_{\mathrm{NM}}$ is the matrix of optimal fitness evaluations. N1 and N2 are the number of individuals of each population.

Algorithm 3 Υ-Metric Pseudocode

```
function UPSM(J_NM, J*_NM, N1, N2)
    ups = ∞. * ones(N, 1)
    for i = 1 : N1 do
        for j = 1 : N2 do
            Ups(i)= MIN(Ups(i), ||J_NM(i, :) − J*_NM(j, :)||)
        end for
    end for
    return AVERAGE(ups)
end function
```

The Δ-metric is computed using the following equation:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}}. \tag{3.17}$$

Equation 3.17 measures spread and evenness of the solutions. The fitness values of the individuals are sorted along one of the objectives to compute the Euclidean distance among subsequent individuals. $\bar{d}$ denotes the average of the differences, d_f and d_l are the distances between the extreme solutions to the extremes of the Pareto front. $\Delta \to 0$ when the distances among the solutions are closer to the average $\bar{d}$. This indicates the solutions are uniformly spread along the non-dominated front. Extremes are computed separately because multi-objective evolutionary algorithms have shown difficulties to find these solutions.

3.3.3 Results

The results are summarized in table 3.1. Sb-MOEA was run for 5000 cycles. Crossover probability was $p_c = 1$ and mutation probability was $p_m = 0.01$. The population had a size of $N = 100$. NSGA-II-R uses a real number codification genetic algorithm. Sb-MOEA obtained lower values of Υ and Δ for the tested problems. Nevertheless NSGA-II-R showed lower standard deviation values. The reference did not report the sample size used to compute their results. On the other hand, 10 tests were conducted to compute the metrics for Sb-MOEA

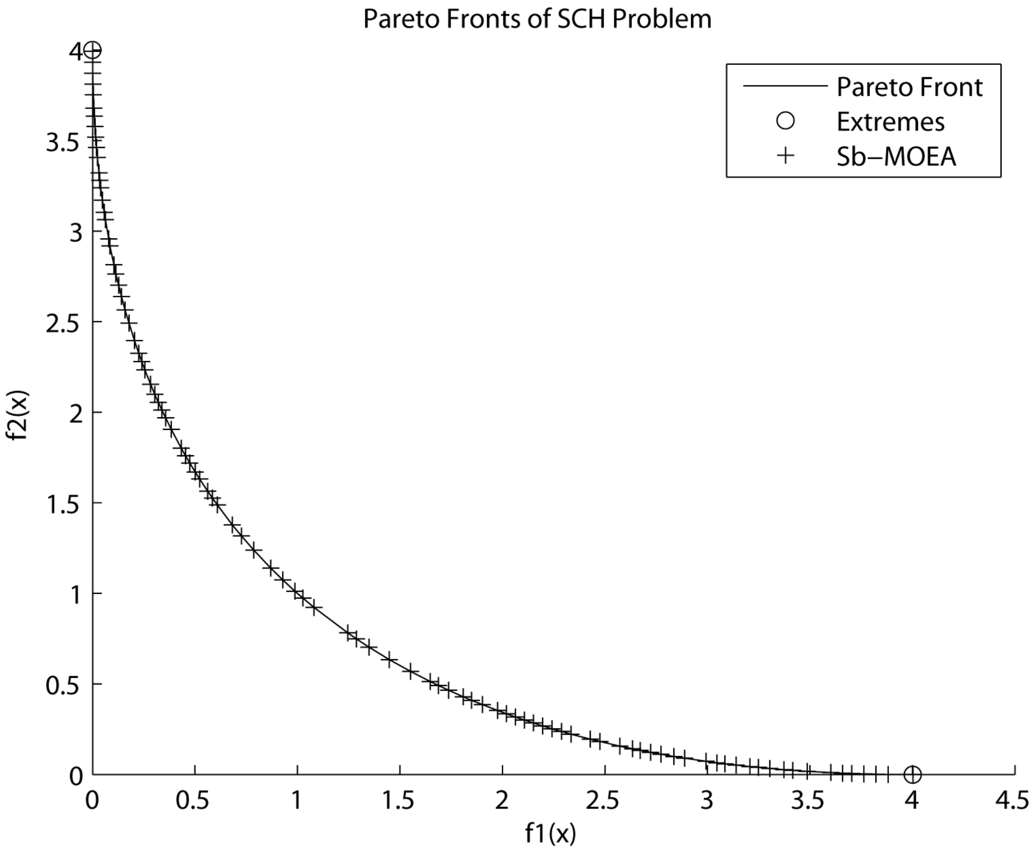

Figure 3.3: Example of Pareto Front computed with Sb-MOEA for SCH Problem.

experiment. Examples of the solutions computed by Sb-MOEA are shown in figure 3.3 and 3.4. The difficulty of the SCH problem consists the Pareto Front region is comparably smaller than the solution space. Usually, the Pareto front spreads along a large region of the solution space, enveloping it. In the SCH problem, more cycles are needed for the algorithm to produce individuals close to the Pareto front region.

The FON problem has the property of being non-convex. This kind of problem was specially difficult for traditional approaches. Sb-MOEA also showed good performance for this problem. A_{ms} was useful during the stage where most individuals of the population are dominated. A_{ms} provides information about closeness to the non-dominated front and spread, which is useful to find more non-dominated individuals. Nevertheless, the critical stage occurs when most individuals are non-dominated. At this point A_{ms} cannot provide further information to guide the process. C_{ms} becomes useful at this stage.

The stochastic fitness mechanism was implemented to stimulate the algorithm to keep looking for non-dominated solutions. $\delta = 0.01$ was used in the experiments. The results seem to indicate a controlled level of noise can improve the performance of multi-objective algorithms. Nevertheless, determination of the required level of noise requires further investigation.

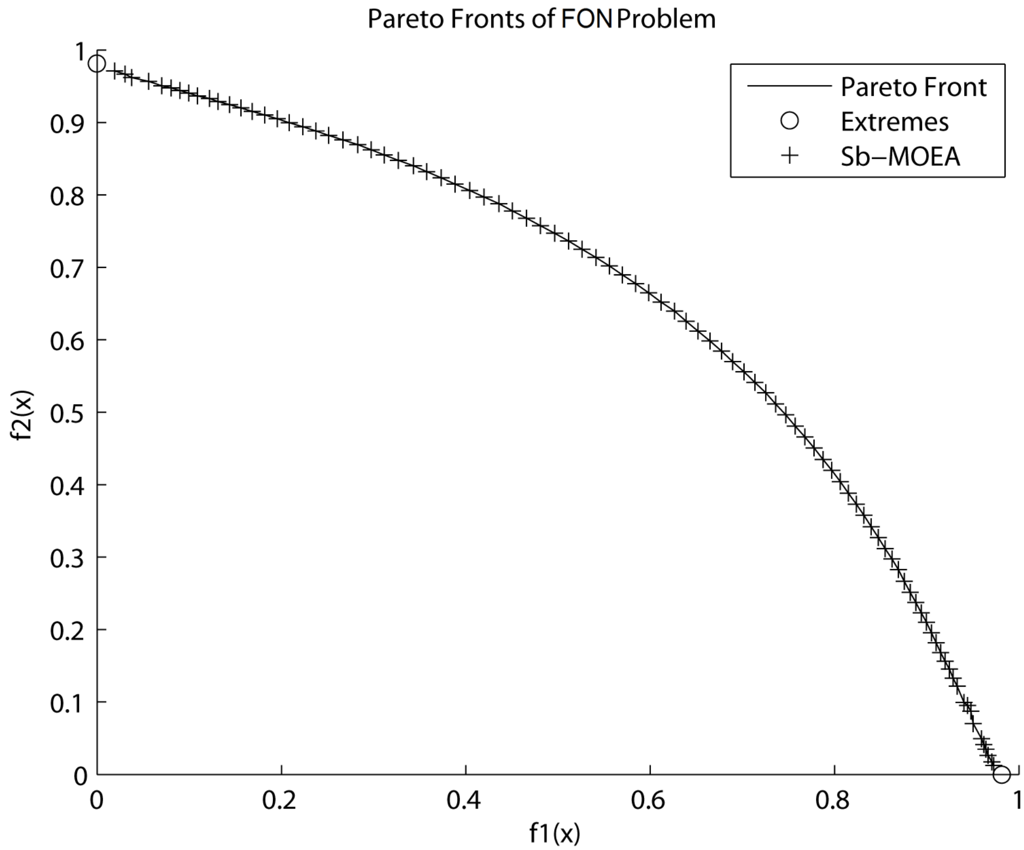

Figure 3.4: Example of Pareto Front computed with Sb-MOEA for FON Problem.

3.3.4 Conclusion

This chapter explained a new multi-objective algorithm to implement the investment method proposed in this work. The literary review showed most of the multi-objective evolutionary algorithms include the following elements: A measure of the closeness of the solutions to the non-dominated front, a measure of density of solutions, and a devise to store the non-dominated solutions. Sb-MOEA implements a new area measure for the first task, the negative ϵ-domination method to promote well-spread solutions, and a non-generational genetic algorithm to store the solutions found. Moreover, the algorithm features a stochastic fitness mechanism to promote further optimization when most of the solutions are already non-dominated.

The new algorithm was tested against results reported in the literature. The chapter explained the test problems and the performance metrics of the experiment. The results indicated Sb-MOEA has good performance when compared with other popular multi-objective algorithms, although, further experimentation should be conducted about this matter. The induction of a controlled noise into the algorithm seemed to be beneficial to find better solutions. Noise was used to overcome the limitations imposed by the assumptions made by density metrics.

SCH	Sb-MOEA	NSGA-II-R	NSGA-II	SPEA	PAES
E[Υ]	0	0.003391	0.002833	0.003403	0.001313
Std. Dev.	0	0	0.000001	0	0.000003
E[Δ]	0.104393	0.477899	0.449265	1.02111	1.063288
Std. Dev.	0.001026	0.003471	0.002062	0.004372	0.002868
FON	Sb-MOEA	NSGA-II-R	NSGA-II	SPEA	PAES
E[Υ]	0.001085	0.001931	0.002571	0.125692	0.151263
Std. Dev.	0.000115	0	0	0.000038	0.000905
E[Δ]	0.123577	0.378065	0.395131	0.792352	1.162528
Std. Dev.	0.001922	0.000639	0.001314	0.005546	0.008945

Table 3.1: Experiment Results for SCH and FON problems.

Chapter 4

Multi-Objective Risk Optimization

The past chapter introduced a new multi-objective algorithm called Sb-MOEA. Although, the algorithm was designed for classical multi-objective optimization. This chapter explains further modifications to Sb-MOEA to solve multi-objective stochastic optimization problems. The enhanced algorithm will be used to solve multi-period portfolio selection problems in later chapters. The proposed method implements a Monte-Carlo approach to overcome the limitations encountered when the problem is defined within a dynamic programming framework. Monte-Carlo methods rely on simulations of many possible scenarios to estimate the probability distribution of the phenomenon. Although, this approach is computationally costly. This chapter proposes some methods to save evaluations. The original structure of Sb-MOEA was considered when devising the proposed methods.

The first part presents a brief review about stochastic optimization. Some evaluations saving methods are derived form the review conclusions. The modified algorithm was tested with a stochastic version of the traveling salesman problem. This problem was selected to provide further information about the properties of the proposed methods from a perspective outside the finance framework. The results are discussed in the last section.

4.1 Stochastic Optimization Review

Uncertainty is a practical problem to be overcome in a wide number of applications. Some examples are incomplete surveys, noisy signals from sensors, unknown future prices of stock markets, and unexpected outcomes from control signals.

The literature about uncertainty and evolutionary algorithms can be classified in two groups: Noisy optimization and robust optimization. References from each type are presented below. The differences from both approaches were identified based on this review. The term *stochastic optimization* is used in this work to refer to any of these terms instinctively. Related concepts will be addressed by this term as well.

Noisy optimization is concerned with finding the best estimation of the optimal solution even when noise is present. Noise is a nuisance to be eliminated in this approach. On the other hand, *robust optimization* admits variations and tolerances are inherent to most problems. Moreover, some of them are out of the control of the designer or they are only partially known. The objective of robust optimization is finding the best solution which is insensitive to these variations. Uncertainties could be understood as restrictions to the original problem. The optimal solution and the optimal robust solution are usually different from each other. Variations on temperature and manufacturing tolerances are some examples of uncertainties.

4.1.1 Noisy Optimization and Evolutionary Algorithms

The study of noisy optimization using genetic algorithms can be traced to early contributions. DeJong (1975) was probably the first one testing genetic algorithms on noisy functions. The results showed genetic algorithms are capable of working properly even when noise was present. The effects of averaging and population size were investigated in later references (Fitzpatrick & Grefenstette, 1988). Zero-mean Gaussian noise was considered in that work. Besides, the problem was solved considering a fixed amount of available evaluations. Different combinations of sample size and population size were compared in the experiments. The conclusion was that a large enough population can cope with noise even when the estimation of fitness used small sample sizes.

Later, the interest of noisy optimization changed to study the performance of genetic algorithms (i.e. convergence speed) in noisy environments (Aizawa & Wah, 1994). The number of evaluations used to solve the problem could be variable, giving room to evaluation saving opportunities. A couple of evaluations saving methods were proposed: The duration-scheduling strategy and the sample allocation strategy. The former changes the number of evaluations per generation while individual sample size is fixed. The latter is the contrary case. Both schemes were tested with problems where Gaussian noise was added to fitness values.

Branke and Schmidt (2003) studied the effect of noise during the execution of genetic algorithms. The conclusion was a low level of noise is not a nuisance because genetic operators purposely add noise to the selection process. This allows the algorithm escaping from local optima. Besides, probabilistic tournament selection was further investigated in that reference. They reported expressions for the number of necessary evaluations to choose correctly the best individual at tournament selection with certain probability. Some other references extended these ideas and devised methods decide the sample size dynamically (Cantú-Paz, 2004).

Also, some references reported the application of multi-objective evolutionary algorithms to solve noisy problems. These references indicate this approach still remains with open questions (Jin & Branke, 2005).

Other works also studied the performance of non-generational multi-objective evolutionary algorithms when solving noisy problems (Hughes, 2000). The non-generational approach was considered greedy for this application because of its high selective pressure. Poorly estimated individuals had a high probability to be unfairly deleted from the population. A probabilistic insertion/cut operator was proposed to alleviate this problem. The operator sorts both parents and offspring and gives deletion probabilities proportional to their ranks. The ranking is also used for the selection process. The operator can be tuned to allow good algorithm performance for a given level of noise in the objective function.

The concept of convex hull has been reported to be useful when measuring the uncertainty of individuals (Trautmann, Mehnen, & Naujoks, 2009). In this approach, the convex hull of an individual is computed from a fixed amount of evaluations. The average distance from the median of measurements to the convex hull determines the uncertainty limit. Domination of individuals beyond this limit can be computed directly. On the contrary, uncertain (i.e. overlapping) individuals are considered non-dominated by default. The NSGA-II was modified to include the convex hull concept. Also, estimation of distribution algorithms (EDAs) has also been applied to solve noisy problems. For example, Shim, Tan, Chia, and Al Mamun (2013) proposed an EDA based on restricted Boltzmann machines for noisy optimization.

4.1.2 Robust Optimization and Evolutionary Algorithms

An early contribution reported the application of evolutionary algorithm to find robust multilayer optical coating designs (Wiesmann, Hammel, & Back, 1998). They considered uncertainties in the thickness of layers. A $(25+50)$-ES was compared to a parallel diffusion model, where a grid is used to determine neighborhoods and compute robustness. The solutions found with the $(25 + 50)$-ES showed higher robustness than the ones found with the other method. Both approaches showed to be computationally expensive, although, they are faster than the traditional design approach.

Robustness can be measured in different ways. For example, Jin and Sendhoff (2003) proposed using the average of the ratio of the standard deviations of the individual's fitness and the standard deviations of the design parameters to measure robustness. These values are computed from a neighborhood around the individual. A single-objective measure is obtained using a dynamic weighted aggregation method. An evolutionary strategy (ES) was implemented to solve some mathematical test problems. The conclusions discussed the possibility of using the proposed method to solve multi-modal robust optimization problems.

Another robustness measures was proposed by Gunawan and Azarm (2005). They introduced a multi-objective approach to solve robust optimization problems along to a worst-case sensitivity region measure. This region represents the hyper-sphere surrounding the evaluated individual and its neighbors. All the neighbors whose fitness values are lower than a threshold of maximum variation are inside the hyper-sphere. A larger radius means the solution is less sensitive to parameters variation. This method was integrated with the multi-objective genetic algorithm (MOGA) and tested with a pinned-pinned sandwich beam design problem. The method was compared against to MOGA without the robustness measure. The proposed method was able to find more robust solutions. An extension of that work considered robustness to be one of the optimization objectives instead of a problem restriction (M. Li, Azarm, & Aute, 2005).

Ong, Nair, and Lum (2006) proposed two methods to measure robustness. The first one is an implicit averaging approach where noise is added to build a neighborhood. The second one generates a set of mutated individuals for the same end. In both cases, the fitness value of the worst-case individual from the neighborhood was used to measure robustness. The methods were tested against a single-evaluation approach. Explicit averaging was also used for comparison. All the measures were implemented using genetic algorithms. The proposed approaches showed better performance and used less evaluations than the other techniques.

Some surveys about the application of evolutionary algorithms of to uncertain environments are available in the literature (Jin & Branke, 2005). In that work, four sources of uncertainty were identified: Noisy objective functions, uncertainty in parameters, uncertainty in the objective function and non-stationarity. Jin and Branke (2005) reported implicit averaging and special selection mechanisms are usually applied to noisy optimization problems. On the other hand, explicit averaging was usually applied to robust optimization problems. Moreover, robust optimization has the advantage disturbances can be deliberately chosen, allowing to have good estimations with smaller sample sizes. Uncertainty in the objective function is usually induced by the application of approximated models. Non-stationarity is usually coped with diversity-generation schemes and memory-based approaches.

D. Lim, Ong, and Lee (2005) presented the called inverse robust design approach to solve multi-objective robust problems. That work claimed robust optimization approaches introduce a prior structure to uncertainty and take advantage of it to solve the problem. A combination of the NSGA-II with sequential quadratic programming was proposed to solve robust problems without any previous assumptions. The expected fitness of individuals is estimated from a set of mutations. Local search is used to identify the worst case, which is used as the robustness measure. The algorithm was tested with mathematical test problems and different parameter configurations. The method was able to find robust optimal solutions but it was regarded as computationally costly.

Paenke, Branke, and Jin (2006) studied the application of Monte-Carlo methods and local models to solve the problem. They were used to estimate the expected quality and variance of the solutions. A regular genetic algorithm was modified to have an archive, and local models were constructed from it. Interpolation and local regression are applied for the estimation. The results showed the method required less evaluations than implicit averaging.

Robust optimization surveys were also found in the references (Beyer & Sendhoff, 2007). Three general approaches were identified in that work: The robust counterpart approach, the aggregation approach, and probabilistic threshold measures. The first one assumes the limits of the parameters variations are fixed. This is a worst-case approach. The second approach assumes the user provides the probability distribution of the uncertain parameters. The probability of the parameter exceeds a particular value is computed directly from the distributions. A Pareto front of solutions can be found with this method. The last approach generates realizations of the performance of a particular design. This approach pretends maximize the number of realizations which performance is below a threshold. Evolutionary algorithms are often used with the third approach but their application are limited because of computational cost.

The problem of reliability optimization was distinguished from robust optimization in the literature. The difference is the solutions should satisfy the uncertainty constrains with a specified probability (Deb et al., 2009). The problems of single-objective multi-modal optimization, and multi-objective reliability optimization were studied in that reference. Different reliability measures were also tested in that work. The results were compared against classical reliability methods. A real-coded genetic algorithm and the NSGA-II were applied to the proposed problems. Mathematical functions were used to test the first two cases, and a car-side impact design problem was used to test multi-objective algorithms. Double-loops methods and the sequential optimization and reliability assessment method (SORA) were used for comparison. The evolutionary algorithms needed less computational effort, compared to the classical methods.

4.1.3 Review Conclusions

The review found other concepts related to optimization under uncertainty besides noisy optimization and robust optimization. Approximated evaluations methods were proposed when the actual evaluations were costly or scarcely available. Noise is artificially induced in the optimization process by these methods. Some noisy optimization applications proposed meta-models to save evaluations in noisy environments. Another related case is non-stationary problems. Both concepts can be referred under the stochastic optimization label.

The references indicate noisy optimization is highly concerned with saving evaluations for the explicit averaging method. On the other hand, robust optimization considers explicit averaging and Monte-Carlo methods to be costly; other approaches have been preferred instead. On the other hand, the literature reported robust optimization approaches have the advantage of defining perturbations and the probability distributions of parameter variation. In this way, a structure is imposed to uncertainty and exploited by the methods to make good estimations with reduced computational costs. Noisy optimization references does not make any assumption about the noise structure. Besides, some references have indicated robust optimization approaches where no assumptions about uncertainty are made is an interesting problem deserving of further attention (D. Lim et al., 2005).

The proposed approach is focused solving multi-objective robust optimization problems where no assumptions are made about uncertainty. This is the case of finance applications, where returns data do not necessarily follow a theoretical distribution. Also, the distribution of final return cannot be known beforehand in the case of multi-period portfolio selection problems.

The chapter is divided in the following sections: Section 4.2 explains the required modifications to the algorithm to perform stochastic optimization. Section 4.3 presents a variation of the traveling salesman problem (TSP) where the time required to travel from a city to another is variable and should be estimated from samples instead of being defined by the user. This problem is used to test the algorithm regardless the framework of finance. Section 4.4 describes the experiments and their results. A Discussion is presented in section 4.5. Section 4.6 presents the conclusions.

4.2 Multi-Objective Risk Optimization Algorithm

In this section, Sb-MOEA is modified to perform multi-objective robust stochastic optimization without uncertainty assumptions. From this point, the term *risk optimization* will be used interchangeably with robust stochastic optimization when referring to this type of problem. The proposed approach is based on a Monte-Carlo method to estimate the probability distribution of the individuals. Although, the literature about robust optimization considered Monte-Carlo methods are too costly. Therefore, this section proposes some methods to save evaluations. The complete algorithm is called structure-based multi-objective risk optimization algorithm (Sb-MORiOA). The methods are explained below.

4.2.1 Evaluations Saving Methods

The proposed approach applies a Monte-Carlo method to estimate the probability distribution of the fitness of individuals. Several realizations are simulated to estimate this distribution, and robustness is measured through its variance. The difficulty of risk optimization problems consists robustness (i.e. variance) estimation requires many more samples than the mean estimation. This can be seen when comparing the variance of the sample mean against variance of the sample variance (Leon-Garcia, 1989). The variance of the sample mean is described by the following equation:

$$\text{Var}\left[M_n\right] = \frac{\sigma_n^2}{n}, \tag{4.1}$$

where M_n is mean estimator, n is the sample size, and σ^2 is the sample variance. On the other hand, the variance of sample variance is

$$\text{Var}\left[\sigma_n^2\right] = \frac{1}{n}\left(\mu_4 - \frac{n-3}{n-1}\sigma^4\right), \tag{4.2}$$

where μ_4 is the fourth moment of the distribution. From these equations, it can be seen that the variance of the sample variance is greater than the variance of the sample mean because the former is proportional to σ^4 while the latter is proportional to σ^2.

Cantú-Paz (2004) proposed a hypothesis testing method to discriminate among two individuals engaged in tournament selection. This method is iterative; individuals are evaluated with a minimal sample size (two samples), then the highest variance individual is progressively evaluated until both individuals pass the test or they reach the maximum evaluations limit. The evaluated individual can change during the process.

In the proposed method, the sample size of the new individuals (i.e. the offspring) is firstly estimated using one of the methods explained below. The actual re-evaluation occurs after the method determines the new sample size. The new sample size is truncated when it exceeds the maximum limit of evaluations per cycle. These methods are based on the comparison of two samples. In this case, the closest non-dominated individual which dominates the offspring (i.e. its dominant) is selected for this end. This one is selected because it is the most critical individual when computing A_{ms}. The closest non-dominated individual is used in the case the offspring is also non-dominated. This one is selected because it is the most critical individual when computing C_{ms}. The method is applied to the offspring only, but re-evaluation of the dominant is recommended to prevent the existence of false non-dominated individuals. Their sample size is increased for a small fraction of the new sample size of the offspring.

In the first method, a t-test is used for the comparison of means and an F-test is used for the comparison of variances. A normal distribution of fitness is assumed in this method. If both tests are positive, no more evaluations are required. On the contrary case, the current mean and variance estimations are kept constant, but the sample size of individuals is increased. The tests are re-calculated for the new set of parameters. The process continues until both confidence values are acceptable or the sample sizes reach their limit. Finally, the individual is evaluated using the sample size obtained by the procedure. The estimations of sample mean and variance are updated accordingly.

An alternative to this method can be found in the references (Diamond, 2001, 2006). This method computes the sample size to guarantee that the risk β of accepting a false hypothesis is small. A modified version used in this work and explained below:

1. Define $\delta = 0.1 \left|\sigma_i^2 - \sigma_j^2\right|$
2. The tests of interest is:
$$H_1 : \sigma^2 \geq \sigma_0^2 + \delta, \tag{4.3}$$
In equation 4.3, σ_0 is the lowest variance and σ is the highest one.
3. Compute the adjustment parameter R as:
$$R = 1 + \frac{\delta}{\sigma_0^2}. \tag{4.4}$$
4. Compute the probability of accepting a false hypothesis C_ν:
$$\beta_\nu = \frac{\chi^2_{1-\alpha,\nu}}{R} \tag{4.5}$$
$$C_\nu = P\left\{\chi_\nu^2 < \beta_\nu\right\} \tag{4.6}$$
5. Compute equation 4.5 until $C_\nu \leq \beta$. Increase ν with each iteration.

The test compares the variances of two different samples. In this case, the comparison is among the variances of two different individuals from the population. Let us assume that the individuals are truly different from each other, but we wish to compute the sample size which acknowledge this fact with a probability of failure equal or lower than β. Step 2 defines the alternate hypothesis of the test, which says the difference among the variances is, at least, δ. This value is defined to be the 10% of the difference of the variances. If the assumption about both individuals are different is true, the sample size needed to confirm it will be relatively small. Although, the sample size will increase when the individuals are closer from each other at the solution space. Step 3 computes the adjustment parameter R, which depends on δ and σ_0. Step 4 computes the critical value β_ν from which the null hypothesis is rejected for a given confidence level α.

The probability of the adjusted value is computed and labeled C_ν. The test is passed if this probability is truly lower than β. The test is computed from the degrees of freedom corresponding to the current sample size N of the tested individual, given $N = \nu + 1$. If the test is not passed, the test is repeated with a higher sample size. The process continues until the test is passed or the maximum sample size is reached. The individual are re-sampled once the new size is determined. Re-sampling is applies to the new individuals from the population.

Recursive Re-sampling Methods

Finally, this section presents recursive methods to compute the sample mean and variance. Their use avoids wasting computational effort because the old estimations of mean and variance are updated but not discarded. These methods can be implemented because of the non-generational scheme of Sb-MOEA. Teknomo (2006) introduced the following recursive equations to compute the sample mean and variance:
$$\bar{x}_t = \frac{t-1}{t}\bar{x}_{t-1} + \frac{x_t}{t}. \tag{4.7}$$

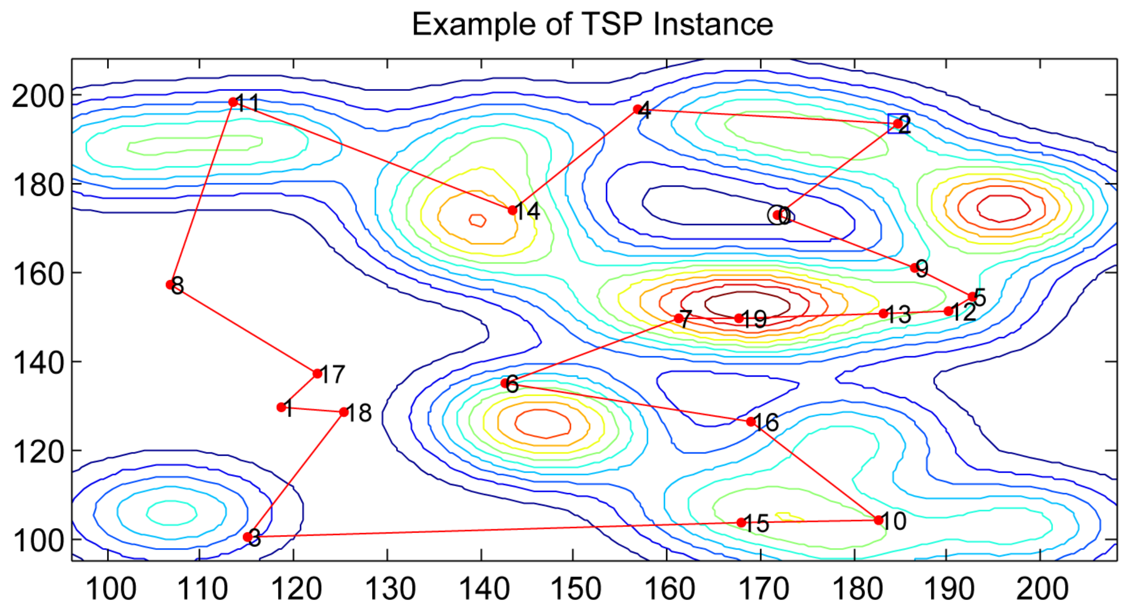

Figure 4.1: Example of TSP Instance.

$$\sigma_t^2 = \frac{t-1}{t}\sigma_{t-1}^2 + \frac{1}{t-1}\left(x_t - \bar{x}_t\right)^2 . \tag{4.8}$$

Where $\bar{x}_t$ is the current average and x_t is an innovation. t is the current number of samples. It increases accordingly until the new sample size is reached.

4.2.2 Custom Modifications

Instances of the traveling salesman problem were used to test the algorithm regardless of the finance framework. Although, further modifications are needed to allow Sb-MORiOA to handle this problem properly. In the proposed method, random keys (RKs) are used to encode permutations (Bean, 1994). Examples about their applications can be found in the literature: Mendes, Gonçalves, and Resende (2009) used them to solve resource constrained allocation problems. Also, Gonçalves and Resende (2011) proposed biased random keys to solve combinatorial optimization problems.

The random keys method encodes individuals using a random number per gene, usually in the range $[0, 1]$. They can be converted to a permutation by simply assigning them a rank in ascending or descending order. For example, the individual (0.3, 0.45 0.91 0.55) encodes the permutation (1,2,4,3) using ascending order. Blocks of bits of arbitrary length can be used to represent random keys with any resolution desired. Crossover of two different individuals will produce a new set of random keys. Therefore, the result will be always a valid permutation. Both crossover and mutation will take place in the random keys space instead of the bits space. In the case of mutation, a new random key is generated for the designated allele.

4.3 Stochastic Traveling Salesman Problem

The traveling salesman problem (TSP) is a widely studied combinatorial optimization problem. A number of cities is represented by a graph. The edges represent the cost (e.g. distance) to move from a city to another. A salesman wishes to find the lowest cost round route which allows him to visit all cities exactly once. The graph is not necessarily complete and the edges are not directed.

There are many variations of this problem. A stochastic version of it is proposed in this chapter. In this version, the edges represent the time needed to travel from a given city to another. A probability distribution of time is associated to each edge but the distribution is unknown to the algorithm. Sampling is needed to estimate the distributions while the optimization process is running. Exhaustive sampling of all the edges seems excessively costly. In this way, the algorithm cannot do any assumptions about the structure of uncertainty, making it a stochastic robust optimization problem. The following method is proposed to generate random instances of TSP:

1. Let us denote the coordinates location of cities by C. p pairs are randomly generated where $C_i = \{c_{xi}, c_{yi}\}$ from $i = 1, 2, \ldots, p$.

2. Let us denote the coordinates location of noise sources by D. q pairs are randomly generated where $D_j = \{d_{xj}, d_{yj}\}$ from $j = 1, 2, \ldots, q$.

3. Each noise source D_j generates a noise field F_j. Fields are modeled as bi-variate normal distributions with the mean vector $[d_{xj}, d_{yj}]$ and identity co-variance matrix $\mathbf{I}$. The mathematical expression is shown below:

$$F_j = \frac{1}{2\pi} e^{-\frac{1}{2\pi}[(x-d_{xj}-(y-d_{yj})]^2}. \tag{4.9}$$

4. The total noise field F_N is the summation of individual fields.

$$F_N = \sum_{j=1}^{q} F_j. \tag{4.10}$$

   The variance σ_{ab} from going to city C_a to city C_b is computed from the integration of F_N along the edge between the cities:

$$\sigma_{ab} = \gamma \int_{C_a}^{C_b} F_N d\ell. \tag{4.11}$$

   Where γ is a scalar scaling factor from $(0, 1)$.

The mean of the edge L_{ab} is the euclidean distance between C_a and C_b. L_{ab} and σ_{ab} are used to generate different realizations of the time needed to travel from city C_a to city C_b. A normal distribution is used for this end.

Equation 4.11 indicates the noise variance is proportional to distance. Long paths are expected to have higher uncertainty, but short paths with high noise are also possible if these paths are close to the noise sources. The noise sources and the cities locations are different

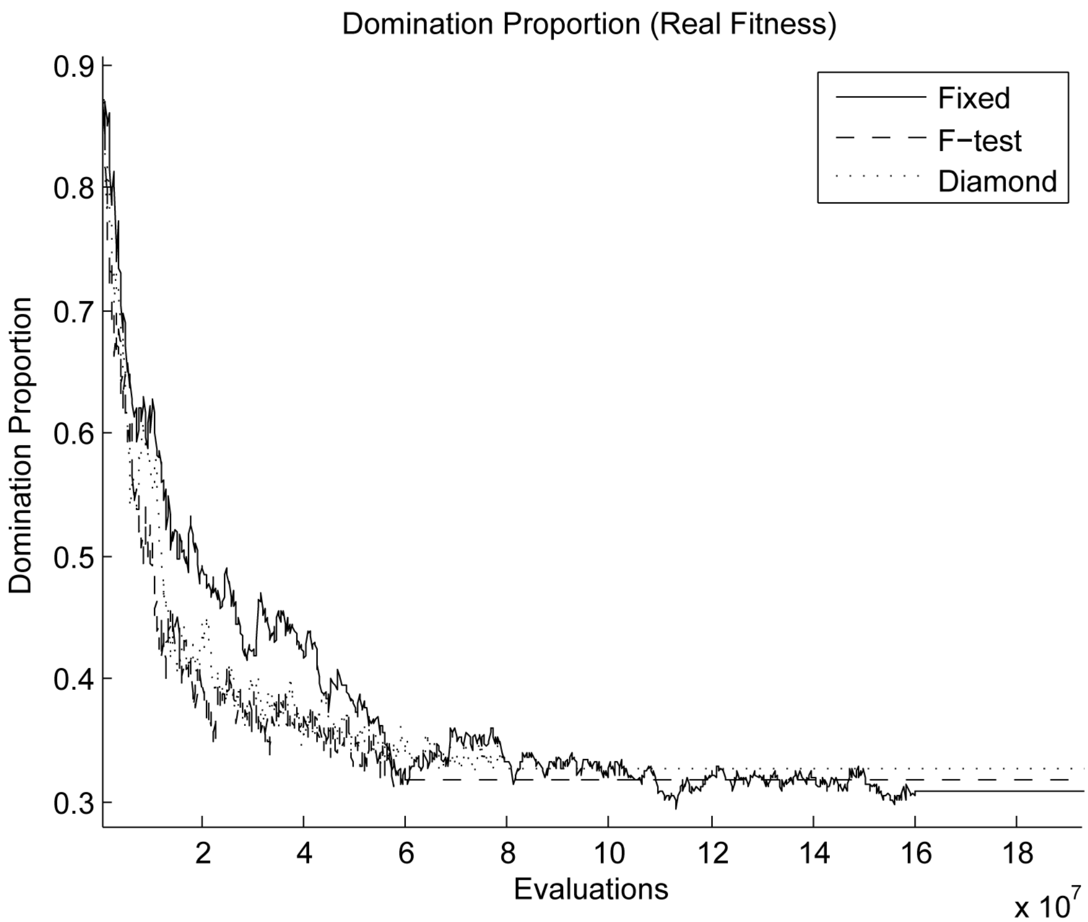

Figure 4.2: Best-so-Far curves for TSP problem instances.

to assure no assumption about the uncertainty can be made from the graph. F_N could be understood as traffic density; more traffic (and delays) are expected in places with a high field density.

Relatively small instances of the traveling salesman problem were used to avoid unnecessary time consumption. Pérez Rave and Jaramillo Álvarez (2013) concluded both small and large sizes of TSP problems are of interest in the current literature. The use of relatively small instances is justified because the presented algorithm optimizes simultaneously several solutions. Figure 4.1 shows an example of a solved instance. The level curves represent the noise field which is used to determine σ_{ij}.

4.4 Experiments

Randomly generated instances of the problem are used for testing. These instances are produced applying the procedure explained in section 4.3. The size of instances is $p = 20$. They

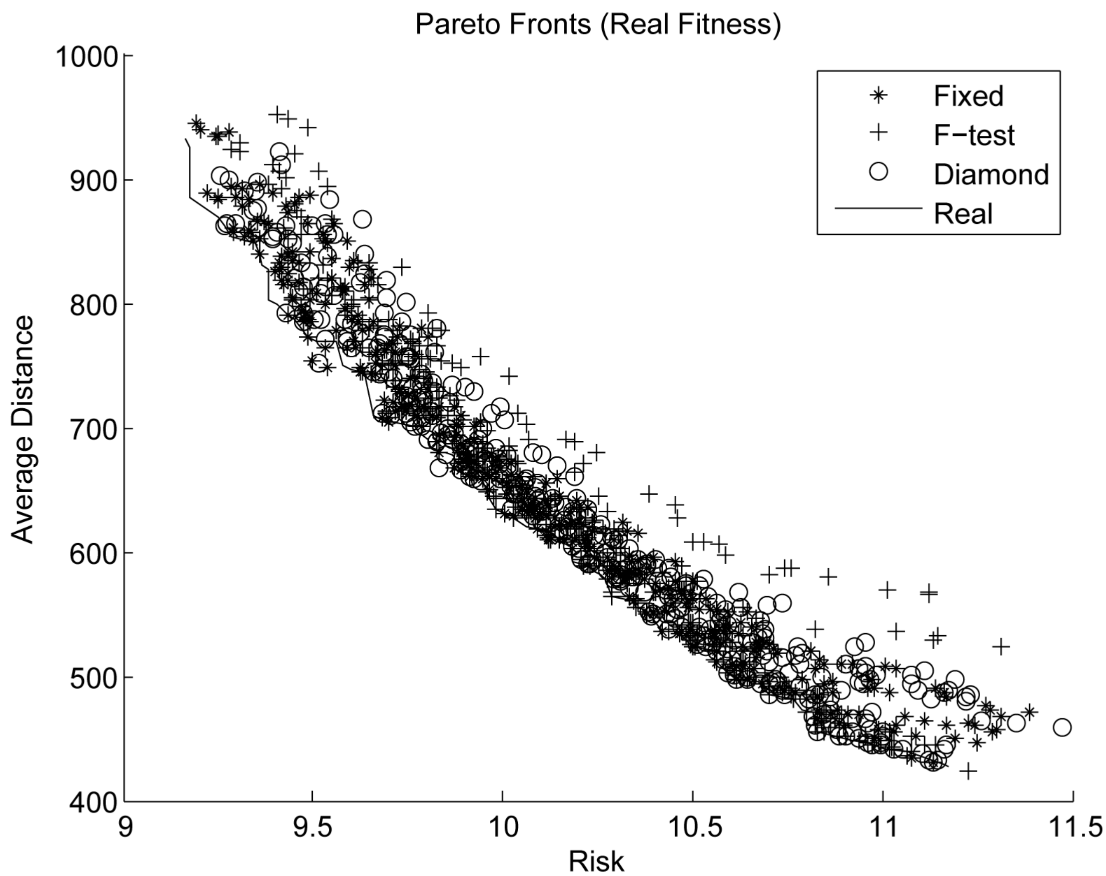

Figure 4.3: Pareto Fronts instances of TSP problem.

Table 4.1: Quality Metrics for TSP Problem Instances.

Instance	Υ metric	Std. Dev.	Δ metric	Std. Dev.
TSP Fixed	0.0052	0.0064	0.4961	0.2340
TSP F-test	0.0134	0.0142	0.5749	0.2851
TSP Diamond	0.0093	0.0117	0.5281	0.2665

are optimized using Sb-MORiOA, and the results are compared against a Sb-MORiOA with a fixed evaluation scheme. Besides, the noiseless Pareto front is estimated and used for the comparison. In this case, the actual values of mean and variance are provided to Sb-MOEA, eliminating the estimation step.

In the experiments, the population size is 50, crossover probability is $p_c = 0.9$, mutation probability is $p_m = 0.1$, The confidence value is $\alpha = 0.95$ and $\beta = 0.01$. The maximum number of evaluations used is 5000. 10% from this value was used to initially estimate fitness values. The algorithm run for 8000 cycles. All instances are complete and not directed graphs.

Figure 4.2 presents the average best-so-far curves for the fixed number of evaluations method, the t/F-test method and the Diamond's method. The best-so-far curve shows the evolution of fitness of the best individual against the number of evaluations along the run. This comparison is difficult in multi-objective algorithms because the Pareto front is composed by many solutions. The number of dominated solutions against evaluations is shown instead. 30 random instances were taken for the experiments, where 10 of them were randomly selected to test one of the three methods. The estimated Pareto fronts were saved to collect information about their quality.

The algorithm works with the estimations of expected fitness and standard deviation, although, actual values of the mean and the standard deviation were used to compute the estimated Pareto front from the final population. The population was mostly non-dominated when estimated values were considered, but some degradation was found when the actual values were used to compute the non-dominated front instead. This difference occurred because the algorithm had no access to the actual values. The non-dominated front of actual fitness values found by the algorithm is the one reported because it provides information about its capacity finding robust solutions under the explained conditions. This one is reported in figure 4.3. The metrics Υ and Δ are used to measure the solution quality (Deb et al., 2002). The non-dominated front of actual values is the one used to compute the metrics of performance. Results are shown in table 4.1.

Finally, an attempt was made to compute the average quality of the non-dominated fronts computed by the algorithm. All the obtained solutions were clustered using the K-means method. The number of clusters was empirically tested to determine a final value $K = 11$. Figure 4.4 shows the clustered average Pareto front computed from clusters for each method. The standard deviation of clusters for the expected time their robustness are presented in figures 4.5 and 4.6, respectively.

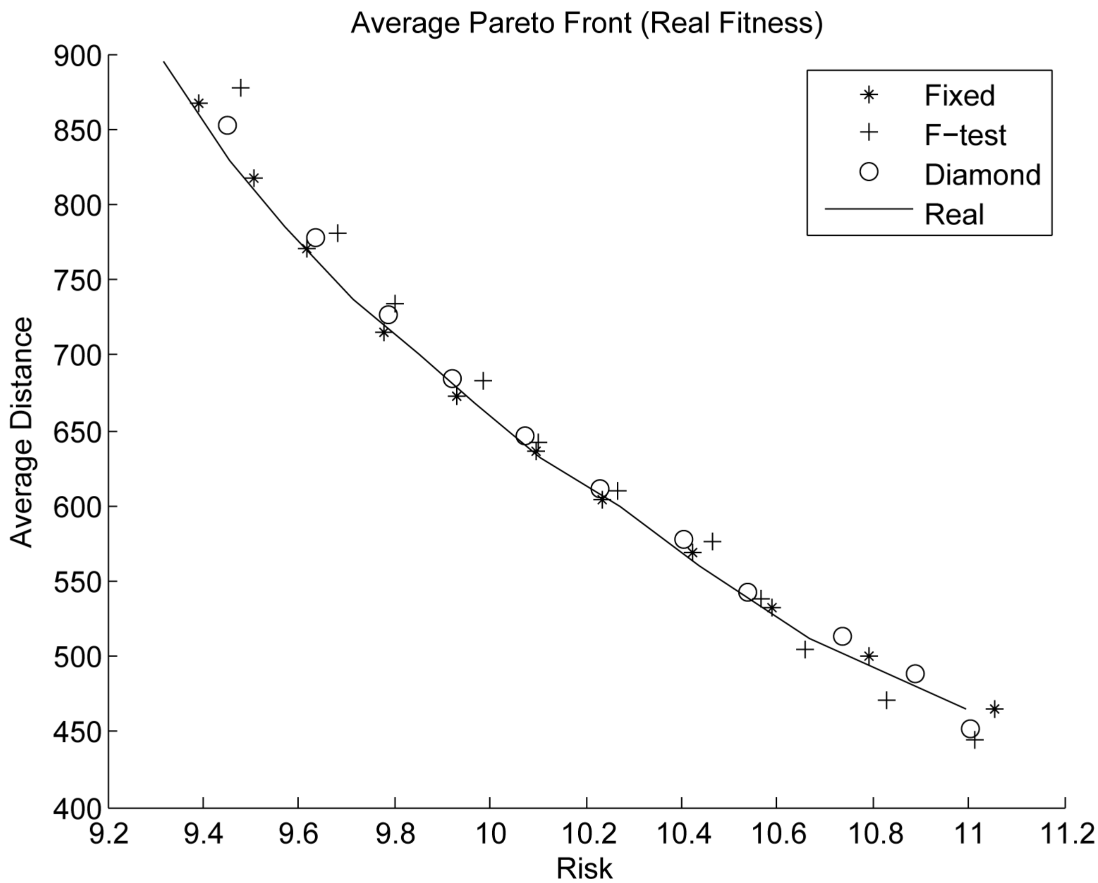

Figure 4.4: Average Clustered Pareto Front of TSP problem.

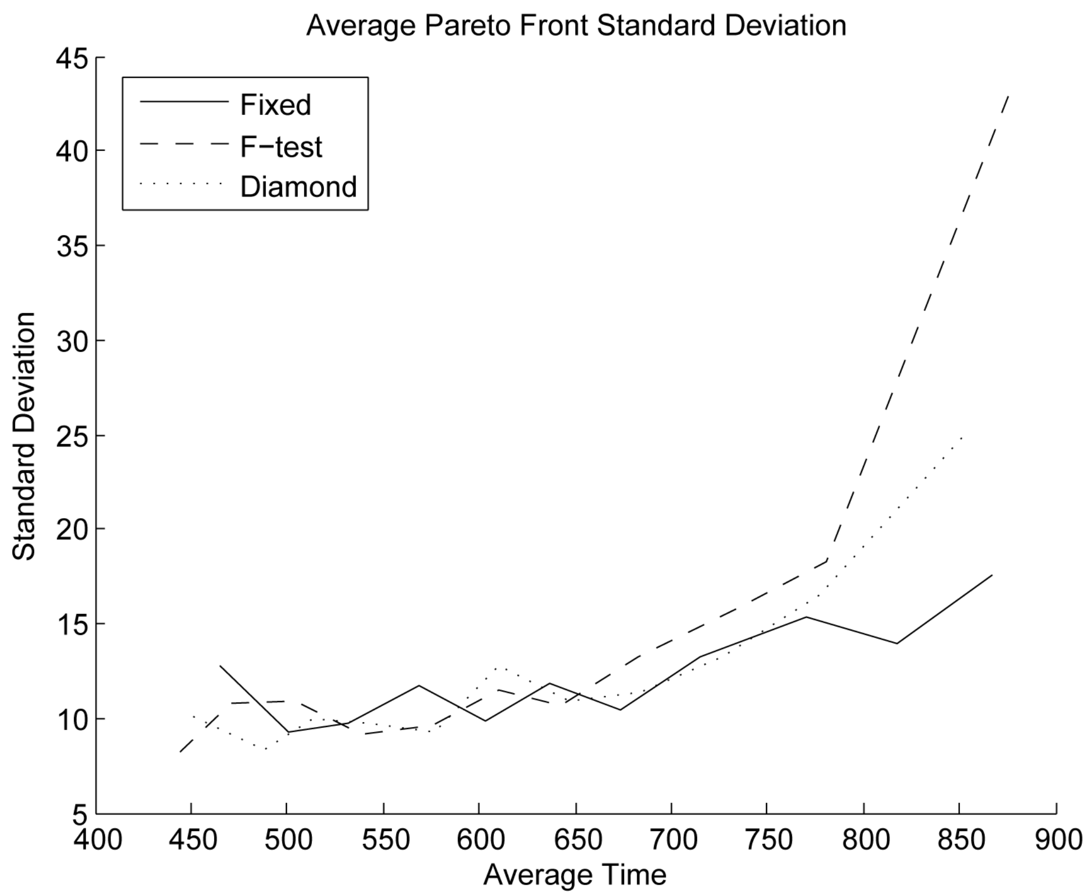

Figure 4.5: Standard Deviation of Clustered Distance for TSP problem.

4.5 Discussion of Results

Table 4.1 shows the average and standard deviation of the quality metrics. Υ-metric computes the closeness of the non-dominated front to the true Pareto front. Δ-metric computes the spread of solutions along the Pareto front. Both measures were computed from the actual values of expected time and its standard deviation, but the algorithm relied on estimations for the optimization. Degradation was found when the actual values were considered. The results show the Diamond's method was the best of the variable methods, although the fixed evaluation method attained slightly better results. On the other hand, figure 4.2 shows the variable evaluation methods are faster than the fixed evaluation method. In this case, the Diamond's method was the one with the lowest proportion of dominated individuals from the two variable methods. The fixed evaluation method attained a slightly lower proportion than the variable methods.

In general, the proportion of dominated individuals was around 30% to 35%. This could be considered a high proportion for multi-objective algorithms. On the other hand, when the estimated values (the ones truly optimized by the algorithm during the run) were considered, the proportion of dominated individuals for the three methods was around 0% to 4%. The latter value corresponds to the case where every individual in the population but the offspring are non-dominated. This result indicates degradation occurs when the actual fitness values are considered when computing the Pareto front from the final population. Degradation is a consequence of the estimation error. There are critical cases where small variation could lead to mistakes when determining domination between individuals. The main limitation of precision is imposed by the maximum limit of allowed evaluations. In this case, the user should make a decision about the trade-off among precision and computational cost.

Figure 4.4 shows the average Pareto front computed for the three methods. This figure is presented to provide more information about the quality of the solutions obtained by the algorithm. The three methods showed similar performance. Moreover, the t/F-test method showed an interesting behavior. The clusters located at low variation regions of the Pareto front attained lower performance than the other methods, but the situation reversed in the high variation region of the Pareto front. This method outperformed the reference Pareto front solution for that section. One cluster for the Diamond's method also outperforms the reference Pareto front. Figures 4.5 and 4.6 show the standard deviation of the clusters for each one of the objectives. In the first figure, the standard deviation of time shows an increment which is proportional to expected time. The t/F-test method is the one with the largest variation. The robustness variation also shows a similar behavior, but the behavior of t/F-test method contradicts the other in this case. These results seems indicate the variable evaluations methods work better than the fixed evaluations methods when higher levels of uncertainty are present. Further investigation is needed to fully understand this phenomenon.

4.6 Conclusion

This chapter presented the modifications to Sb-MOEA to perform stochastic multi-objective optimization without assumptions about the uncertainty and the inclusion of Monte-Carlo methods. Evaluation saving methods were devised based on the structure of Sb-MOEA. The

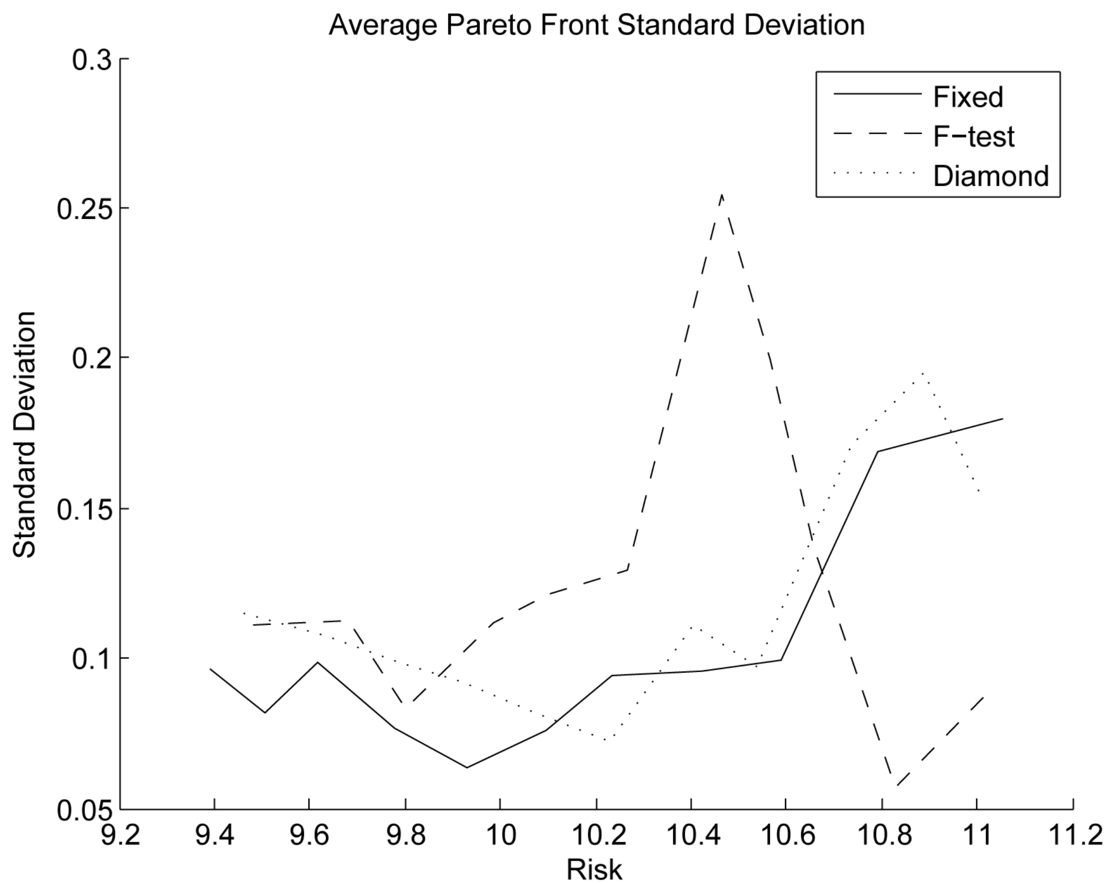

Figure 4.6: Standard Deviation of Clustered Risk for TSP problem.

complete algorithm was named Sb-MORiOA. This algorithm was devised to implement the solution model to multi-period portfolio selection problem explained in later chapters. The evaluation saving methods are mainly based on hypothesis testing techniques.

The algorithm performance was studied using a stochastic traveling salesman problem to analyze their properties regardless the finance framework. Also, it discusses a method to generate random stochastic instances. The evaluation saving methods were compared against a Sb-MORiOA with a fixed evaluation scheme. The actual Pareto front was estimated using Sb-MOEA. The actual values of mean and variance were provided to the algorithm for this end.

The results show the saving evaluation methods are able to attain similar solution quality than the fixed evaluation methods with less computational cost. Sb-MORiOA with variable evaluation methods was able to outperform Sb-MOEA with actual fitness values for high uncertainty regions of the Pareto front. Moreover, other saving evaluations methods can be devised and tested with different stochastic problems.

Chapter 5

Solution Model

The objective of this work is to develop an evolutionary computing method for financial decision making. A literary review was conducted and the conclusion was a multi-period portfolio allocation framework was suitable for this end. That is because the multi-period framework allows the inclusion of portfolio selection and diversification. Also, it allows the inclusion of dynamic effects like transaction costs, portfolio unbalance, and state-dependent investor's preferences. A Monte-Carlo approach was suggested in the literature to overcome the difficulties to solve this problem analytically. The algorithm was fully explained in past chapters. Nevertheless, the method to apply this algorithm to solve multi-period portfolio selection problems has not been addressed yet.

This chapter explains the solution model to solve the problem and perform financial decision making. The Monte-Carlo method is explained along with some modifications to Sb-MORiOA to perform portfolio optimization. This method use multi-period Pareto fronts for dynamic portfolio allocation with parametric inclusion of the investor's preference.

5.1 Multi-Period Portfolio Selection Model

The references showed portfolio selection has been widely investigated, both theoretical studies and applications can be found. Multi-period approaches differ from single-period ones when the investor's utility function is state-dependent. Transaction costs affect utility functions because they penalize state changes, and inflation makes utility time-dependent. Unbalance occurs naturally when a portfolio is kept unrevised for some time, changing the state of the portfolio. The multi-period portfolio selection problem can be stated as a dynamic programming problem, but its solution has proved to be difficult. Theoretical works usually find a closed-form optimal solution for some particular conditions. There were reported varied applications using evolutionary algorithms, but multi-objective algorithms seem popular because the efficient frontier can be computed in a single run. Cardinality and minimum lot restrictions have been considered in some reports, which differs from multi-period approaches because they deal mainly with transaction costs and state-dependency. Single-period approaches are distinguished from multi-period ones. Multi-period approaches generally deal with optimization of final utility and risk. Monte-Carlo approaches have been used to allow to solve the dynamic programming problem and obtain numerical solutions instead of closed-form solutions.

Single-period approaches can deal with data innovations directly, but their solutions are not optimal when state-dependent factors are considered. On the other hand, multi-period approaches deal naturally with state-dependency, but innovation data is not directly integrated in the solution. The suggested approach shows a multi-period method which is able to incorporate data innovations, besides the state-dependent factors.

5.1.1 Portfolios Sets Wealth Model

Equation 5.1 shows the model of portfolio wealth $x(t)$. This is a stochastic differences equation where the portfolio's wealth depends on the current return of securities and the past portfolio. The following equation model the portfolio's value:

$$x(t) = \mathbf{r}_1(t)\mathbf{w}(t)^T \left(1 - \text{cost}(t)\right) x(t-1). \tag{5.1}$$

Where cost(t) depends on the current unbalanced portfolio:

$$\text{cost}(t) = c \left\| \mathbf{w}(t) - \frac{\mathbf{r}_1(t-1) \odot \mathbf{w}(t-1)}{\mathbf{r}_1(t-1)\mathbf{w}(t-1)^T} \right\|_1 . \tag{5.2}$$

Equation 5.1 expresses the current wealth $x(t)$ to be a function of present return, portfolio, and the wealth of the past period. A *state* is described by the current portfolio and its wealth. The model considers three dynamic restrictions: Transaction costs, unbalance, and inflation. $\mathbf{r}_1(t) = 1 + \mathbf{r}(t)$ when traditional returns are assumed. On the other hand, $\mathbf{r}_1(t) = \exp(\mathbf{r}(t))$ when logarithmic returns are considered instead. The exponential operator is applied in an element-wise fashion to the components of $\mathbf{r}(t)$.

Transaction costs are fees paid to brokers for each operation. The available operations are the sale or bought of securities. Fees are usually a fraction of the total price of the considered security. Equation 5.1 uses the parameter c to include the transaction cost rate, regardless its type. The limits of this parameter are to be $c = [0, 1]$. Portfolio costs are directly proportional to $\|\mathbf{w}(t) - \mathbf{w}(t-1)\|_1$, where $\|\cdot\|_1$ is the 1-norm of the changes in the portfolio.

Unbalance is the natural change in the portfolio composition due to differences in the return rates of assets. For example, if a 2-assets portfolio has the same amount of wealth invested in each one of them, then portfolio weights can be expressed by the vector $[0.5, 0.5]$. Let us suppose the returns for each assets are $[0.9, 1.2]$ and total wealth to be 1. The amount of wealth of the first security will decrease, and the second one, will increase to be $[0.45, 0.6]$. The new portfolio weights are $[0.4286, 0.5714]$. Unbalanced portfolios tends to accumulate wealth in the most profitable security. The changes in portfolio composition affect both risk and return. Financial professionals usually recommend portfolio revision to correct unbalance and keep risk under desired levels (Sharpe et al., 1999).

Unbalance is the cause the current portfolio to be different from the one initially selected. Unbalance impacts directly in portfolio costs. Equation 5.2 indicates portfolio changes depend on the current portfolio $\mathbf{w}(t)$, the past portfolio $\mathbf{w}(t-1)$, and the past securities returns $\mathbf{r}_1(t-1)$. The operator $\odot$ indicates element-wise matrix product. Transactions costs are directly proportional of the difference of past and current portfolios.

Inflation is included into the model through the interest rate. This concept comes from the fact future wealth has less utility than current wealth. Loaners usually demand a percentage of the borrowed money as payment for the loan. This amount is expressed in the form of the interest rate. This change in utility should be considered when cash flows from different times are compared. The concept of present value (PV) allows this comparison. Any future cash flow $x(t)$ can be expressed in current terms at a given interest rate I as

$$x(0) = \frac{x(t)}{(1+I)^t}. \tag{5.3}$$

I can is also useful to capture the investor preferences. For example, when the final return is less than the inflation rate, this will be perceived as a loss for the investor. I should be adjusted to the time window considered. For example, weekly inflation rate should be used when working with weekly security prices, otherwise, the investor's expectation could be unattainable. Equation 5.1 can be modified to consider investor expectations of the future in the following manner:

$$x(t) = \frac{\mathbf{r}_1(t)\mathbf{w}(t)^T}{(1+I)^t}\left(1 - \text{cost}(t)\right) x(t-1). \tag{5.4}$$

5.1.2 Monte-Carlo Approach

Equation 5.1 is well suited for the application of Monte-Carlo methods. It can be computed iteratively to obtain the portfolio wealth at any time desired. Portfolio can be changed at any time also. Besides, equation 5.1 obtains wealth values for $t = [1, 2, \ldots, T]$, allowing to use this information for decision making.

The solution to multi-period portfolio selection problems is the probability distribution of portfolio wealth x_T. Besides, the estimation of the second or higher distribution moments are necessary when computing risk. Risk should be accurately estimated because is one of the objectives to be optimized. Finding closed-form expression for the expected value and variance of equation 5.4 can be difficult due the effects of transaction costs, unbalance and the investor's expectations. The probability distribution of x_T is unknown even under the assumption securities returns are normally distributed. The Monte-Carlo approach consists generating multiple realizations of returns and use them to obtain samples of the final portfolio return. Probability distribution moments can be estimated from the samples. On the other hand, traditional portfolio theory computes portfolio return and portfolio risk from equations 1.6 and 1.7, respectively.

For the Monte-Carlo approach, historical data are used to generate the necessary realizations. In this case, the data can be used to generate random numbers with the same distribution of security returns and apply them to compute multiple realizations of x_T. The present work assumes returns can be modeled with a multi-variate normal distribution which can be computed from historical data. In other words

$$\mathbf{r}_t \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right). \tag{5.5}$$

Nevertheless, the proposed approach is not restricted to any particular distribution. In equation 5.5, $\boldsymbol{\mu}$ is the vector of the average returns of securities, and $\boldsymbol{\Sigma}$ is the co-variance matrix of securities.

Traditional portfolio theory considers the case where a fixed-income financial instrument is included in the portfolio. Government bonds are an example of them. Fixed-income assets can be included into the model by using their return values directly instead of random generated numbers. Moreover, the ability to liquify the portfolio can be modeled by a fixed-income asset with zero return rate. The current study will be limited to securities and fixed-income assets, but the model could be extended to include other types of financial instruments.

5.1.3 Inclusion of Data Innovations

References about multi-period portfolio problems propose methods finding a set of portfolios which optimize utility of the last period. These approaches are not concerned with the utility from other periods besides the last one. The possibility of decision changing becomes important when data innovations are included. The existence of portfolios sets of different time duration would allow a better control of risk and return; they would provide a wider set of options to adapt to the current portfolio state caused by changes in the market. This work investigated this conjecture. The cited references do not realize the following possibility: Portfolios sets with shorter duration than the time horizon T can also be Pareto-optimal. These portfolios could be plotted along the current multi-period Pareto Front and they will be non-dominated. In that case, they should also be included in the multi-period efficient frontier.

The hypothesis was investigated using a regular genetic algorithm. Equation 5.4 can be implemented to be the objective function. Traditional portfolio theory use a weighted sum approach to find the optimal portfolios (H. Markowitz, 1952). Therefore, the objective function used in this experiment is the following:

$$f_q = kE\left[x_T\right] + (1-k)\sigma_T. \tag{5.6}$$

In equation 5.6, f_q represents the fitness value of the q-th individual. x_T and σ_T represent the wealth and risk values for the portfolio encoded by chromosome q. k is a risk weighting factor where $k \in [0,1]$. The extreme values of k eliminate the effect of one of the objectives. Running the algorithm with combinations of $t = [1,T]$ and $k = [0,1]$ will lead to obtain the multi-period Pareto fronts for different time horizons. If the initial assumption is correct, the efficient frontier will be composed by solutions with different number of periods.

A genetic algorithm was implemented to solve an instance of this problem. Weekly price data from some components of the Dow Jones industrial average were chosen for the experiment. The securities are the following: American Express Company (AXP), Cisco Systems Inc. (CSCO), and Chevron Corporation (CVX), from years 2000 to 2012. The maximum number of periods $T = 5$. The value I was chosen to be the average return of the less risky security. Initial wealth is 10000. Each individual was evaluated 10000 times to compute its expected portfolio value and risk. The result is shown in figure 5.1.

Figure 5.1 shows the Pareto fronts obtained for each time period. The multi-period Pareto fronts for different time horizons overlap with each other. Besides, it shows how some non-dominated solutions become dominated because of the presence of the other Pareto-fronts. The final multi-period Pareto front will be composed by solutions with different time duration. In that case, the investor could retract from his initial period selection and choose portfolios with shorter horizons instead. This possibility is overlooked when the last-period Pareto front is the only one considered. Also, this is a proof that multi-period solutions are different from single-period ones. If they were equal, the Pareto front from different time periods would be unable to dominate solutions from the final time Pareto front. From this point, the term *multi-period Pareto front* will be used to refer to efficient frontier composed by solutions with different time duration.

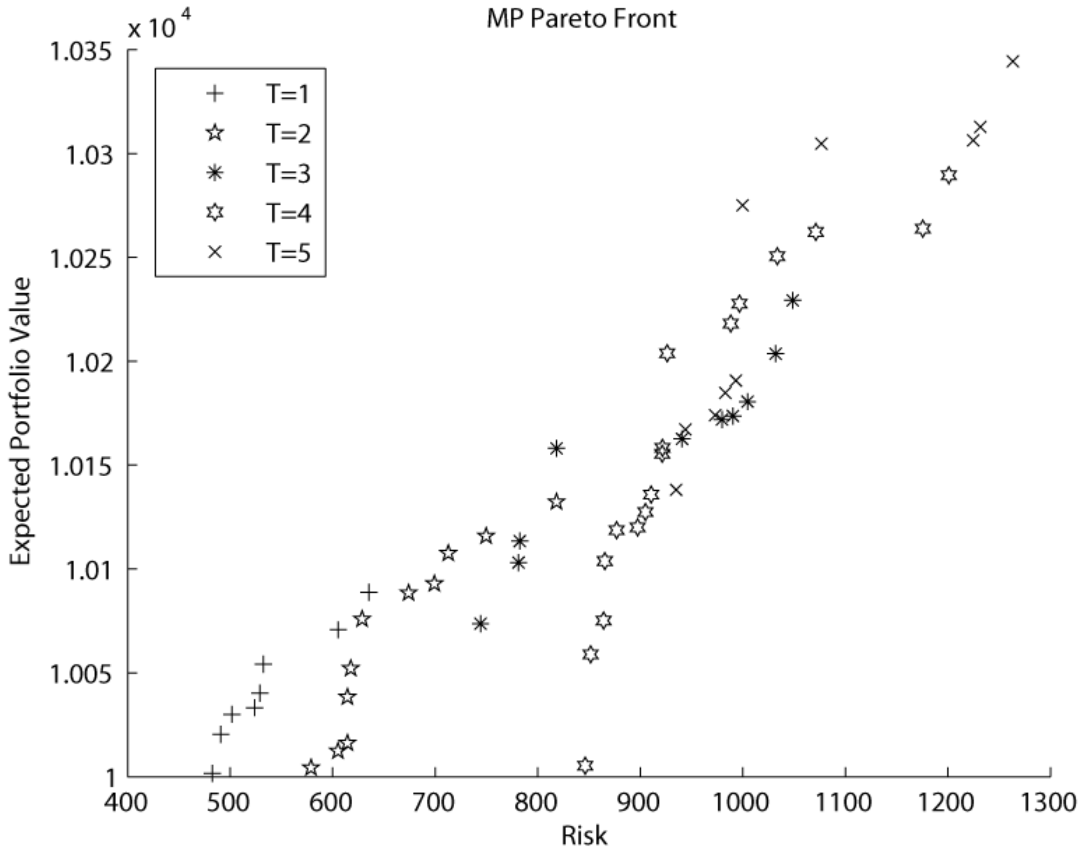

Figure 5.1: Pareto fronts for DJI experiment and maximum period length $T = 5$. Pareto fronts from $T = 1$ to $T = 5$ are shown.

Selection of Portfolios Sets

The proposed method computes multi-period non-dominated fronts where the optimal sets of portfolios may have different number of periods from each other. For example, a set composed by a single portfolio may appear at the same multi-period non-dominated front than a set of 5 periods if both of them are non-dominated. The multi-period non-dominated front is computed from the non-dominated fronts for each period of interest by discarding the dominated solutions when all of them are considered together.

Algorithm 5.1.3 computes the individual non-dominated fronts iteratively using the SB-MORIOA( ) function. We assume the investor is available to provide the algorithm information about his preference when some evidence is presented to him. The multi-Period non-dominated front is computed from the individual non-dominated fronts through the functions UNIFY_PFs( ) and GET_PARETO_FRONT( ). The risk of the highest return set of portfolios is obtained using the function GET_MAX_RISK( ) and labeled σ_M. In the following step, the fraction of the maximum risk $\alpha_{\sigma M}$ is compute with the equation

$$\alpha_{\sigma M} = \frac{\sigma^*}{\sigma_M}, \tag{5.7}$$

where σ^* is the maximum risk tolerated by the investor. Equation 5.7 is implemented in the COMPUTE_ALPHA_SIGMA_MAX( ) function. Investors may have not a clear idea about their preference about risk, but they can make an estimation when evidence is provided. In this case, T and σ^* are the data provided by the investor to indicate his preference. In the algorithm, $\alpha_{\sigma M} > 1$ indicates the investor is still willing to take higher risk for the sake of profit, therefore, more individual non-dominated fronts with larger number of periods should be integrated to the multi-period non-dominated front. On the contrary, $\alpha_{\sigma M} < 1$ implies the preferred level of risk of the investor is attainable for the current multi-period Pareto front and no more fronts are necessary. The set of portfolios which risk is the closest to $\alpha_{\sigma M}\sigma_M$ is chosen as the first investment decision. This set is computed using the function GET_PORT( ). The return of this set of portfolios is labeled x_{Tm}. The other returned variables by the function GET-MPPF( ) are used later for decision making.

Algorithm 4 Pseudocode to Choose Initial Multi-Period Pareto Front and Investor's Expectations

```
function GET-MPPF(N,M,Ev_min,Ev_max,atfobj,Ncyles,w_0,x_0,c,I,data)
    T ← 1
    α_σM ← 1
    MPparF ← φ
    while α_σM ≥ 1 do
        fobj = CONFIG_FOBJ(atfobj,w_0,x_0,c,I,T,data)
        pop = SB-MORIOA(N,M,Ev_min,Ev_max,fobj,Ncyles)
        parF = GET_PARETO_FRONT(pop)
        MPparF = UNIFY_PFS(MPparF,parF)
        σ_M = GET_MAX_RISK(MPparF)
        α_σM = COMPUTE_ALPHA_SIGMA_MAX(σ_M,T)
        T ← T + 1
    end while
    x_Tm=GET_X_T_MIN(MPparF,σ_M,α_σM)
    port=GET_PORT(MPparF,x_Tm)
    return port,MPparF,σ_M, α_σM,x_Tm
end function
```

Effect of Data Innovations

Data innovations are used to evaluate the current state of the portfolio.The final wealth x_T can be computed using equation 5.4 with real-data returns instead of random-generated numbers. Evaluation of equation 5.4 also obtains the current portfolio w_T, which suffered unbalancing during the investment time. Figure 5.2 shows the effect of data innovations in the multi-period

Pareto front. The solid line represents a multi-period Pareto front computed from an out-of-market scenario (i.e. the initial state). This means $x_t = x_0$ and no portfolio is selected. The dashed lines represent multi-period Pareto fronts where $x'_0 = x_T$ and $w'_0 = w_T$. x_T and w_T are obtained when a portfolios set was chosen from the solid-line multi-period Pareto front and evaluated with data innovations. The dashed-line multi-period Pareto fronts are conditioned to the last state of the chosen portfolios set. The conditioned multi-period Pareto front starts above the original multi-period Pareto front when $w_T > w_0$. The contrary occurs when the selected portfolios set suffered a loss.

Investment Strategies (ISs)

There is the possibility the actual value of the portfolio $x_T < x_{\text{Tmin}}$. In this case, further actions are required to satisfy the investor's goals. Besides, there is not need to stop even when $x_T \geq x_{\text{Tmin}}$ because further profit can be obtained. This assumption is in accordance with the insatiability principle. Therefore, a method is needed to continuously select the next set of portfolios. This work calls these rules *Investment Strategies*. ISs are defined by the following equation:

$$\mathbf{S}_t = [s_1, s_2, \ldots, s_M], s_i \in [0, 1] \tag{5.8}$$

Equation 5.8 allows implement different criteria to choose the next set of portfolios. The procedure described by algorithm 5.1.3 is only one of the possibilities. There are two possible scenarios when the multi-period Pareto front conditioned to current state is computed: The new Pareto front dominates the original selection or the new one is dominated by it. This is illustrated in figure 5.2. The solid-line Pareto front is the one computed with entering to market conditions. Portfolios set P was initially chosen. The dashed-line Pareto fronts are computed for scenarios of profit and loss, respectively. The dashed vertical and horizontal lines indicate the risk and final wealth expected by investor. The profitable scenario Pareto front has most of its portfolios above the minimum expected wealth x_{Tmin}. The loss scenario Pareto Front has few portfolios with higher return than x_{Tmin}, and this value can only be reached at higher risk levels than $\alpha_\sigma \sigma_{max}$. ISs should decide the next sets of portfolios according with the current conditions and the investor's goals. For example, for the loss scenario, A is the portfolio with lower risk that reaches x_{Tmin}, but C is the portfolio which risk level is equal to investor initial preference. The portfolio with better balance of both objectives is B. For the profit scenario, E is the highest risk portfolio while D is the most conservative one. E is also the closest one to original investor expectations.

This example shows different criteria could be proposed to choose the next set of portfolios. Different strategies can be implemented by changing these criteria. One of the research questions proposed in the first chapter was about the effect of different ISs and compare them with human behavior. It wold be interesting to determine if counter-intuitive criteria could lead to good ISs.

Algorithm 5.1.3 describes the method to implement ISs. The goal $[x_{\text{Tmin}}, \alpha_\sigma \sigma_{max}]$is the point of reference, computed using algorithm 5.1.3. Algorithm 5.1.3 chooses the set of portfolios with the closest weighted distance to the objective. ISs are expressed in the form of the vector of weights S_t. The rule is applied to find portfolios sets iteratively. This algorithm can be applied while enough innovations are available.

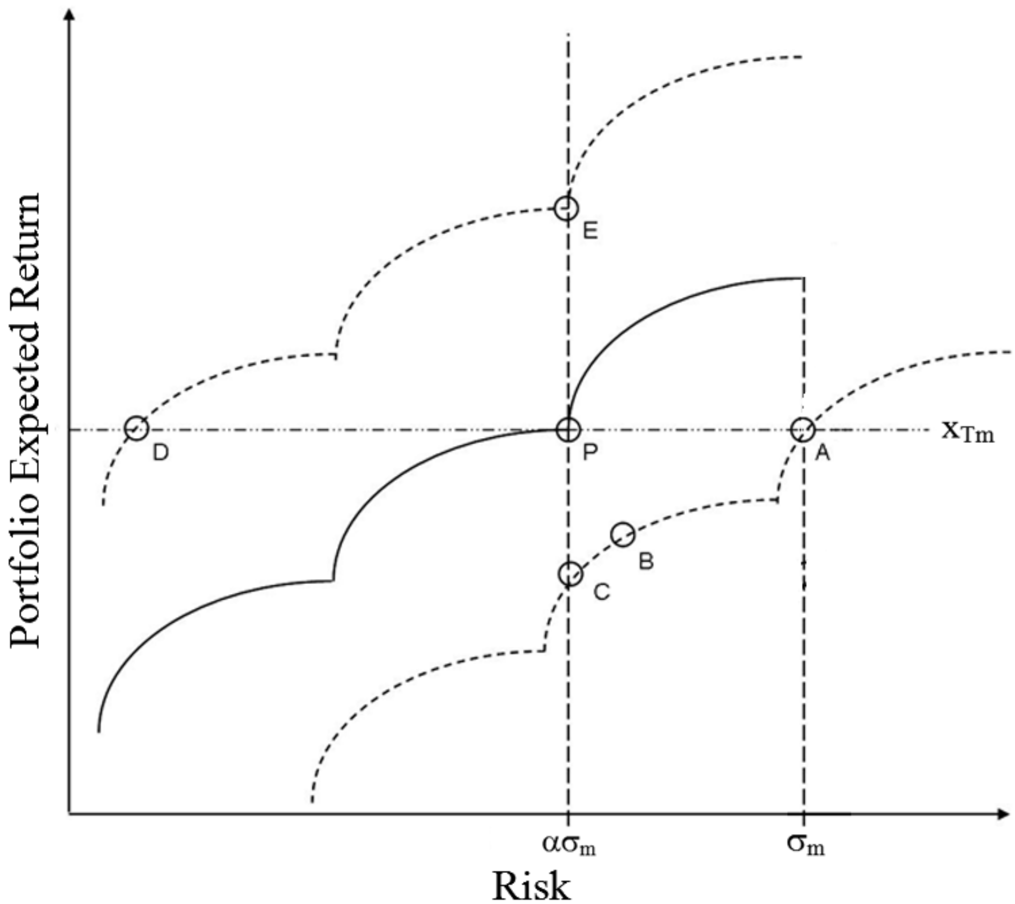

Figure 5.2: MP Pareto fronts. The solid front is computed for coming into market conditions. Dashed fronts illustrate the cases when the decisions were profitable and when decisions caused a loss, respectively. Point P represents the initial portfolios set chosen by the investor.

5.1.4 Selection of Suitable Securities

Fundamental analysis is concerned with the pricing of securities. This approach uses information from the company's financial reports and macro-economic models to estimate their true values. Careful pricing estimation allows the identification of these profit opportunities. The possession of the most promising securities is part of a good investment decision method.

The selection of securities is important in the framework of the present work because computational effort increases with the number of securities optimized. For example, portfolios from the Standard & Poor index (S&P500) will be composed by 500 securities. This number of assets is impractical for most of the investors. The difficulty to replicate market indexes portfolios is discussed in later chapters. The conclusion is a method is needed to select the securities to be considered by the algorithm.

The proposed method consists in computing the expected return and the standard deviations of the securities from historic data. The Pareto front of risk-return is computed from them. The non-dominated securities will be the ones considered by the method explained in

Algorithm 5 Pseudocode to Implement Investment Strategies

```
function APPLY_IS(S_t,PF,N,P,x_Tmin,α_σm σ_m,data,innovations)
    [x_T, Ri_T] = EVALUATE_PORTFOLIO_SET(P,innovations)
    ix ← 0
    mindist ← ∞
    for j = 1 : N do
        [x_j,Ri_j] = OBJECTIVE_FUNCTION(PF(j, :),data)
        dist = ‖([x_T, Ri_T] − [x_j, Ri_j]) ⊙ S_t‖_2
        if dist ≤ mindist then
            mindist = MIN(mindist,dist)
            ix ← j
        end if
    end for
    return PF(ix, :)
end function
```

this chapter.

An experiment was conducted to test the validity of this procedure. It considered all the securities from the Dow Jones average index at 2012. History was collected from 2005 to 2012. The Critical Line Algorithm (CLC) was used to compute the efficient frontier (Sharpe et al., 1999). Figure 5.3 shows the Pareto fronts obtained when the complete set and the reduced set were considered. The solid-line Pareto front was computed using all 30 securities, while the dashed-line Pareto front was computed using the non-dominated securities only. Both Pareto fronts are similar. The difference appear at the low-risk region, where the Pareto front of all the securities dominates the other one. Besides, Pareto front of non-dominated assets does not reach some low risk regions the other one does. Nevertheless, the difference in return and risk is small and the region not reached by the latter Pareto front has negative expected returns. It is improbable the investor chooses one of the portfolios with negative expected return even when they are non-dominated. Therefore, it seems reasonable to apply the investment strategies method to non-dominated securities only.

5.2 Sb-MORiOA Custom Modifications

Sb-MORiOA was tested against instances of the traveling salesman problem in past chapters. A random keys coding was used to allow the algorithm to handle permutations effectively. Portfolios are not permutations, but special considerations should be taken to improve the efficiency of the algorithm. Two modifications are proposed for Sb-MORiOA when handling portfolios: Vector coding and buy-and-hold coding. They are explained below.

5.2.1 Vector Coding

The simplest way to represent a portfolio of M assets in binary form is using M groups of bits to encode each weight. Assuming we are using b bits per gene in the chromosome, bM bits would be necessary to represent the whole portfolio. Denoting g_i to be the decimal value

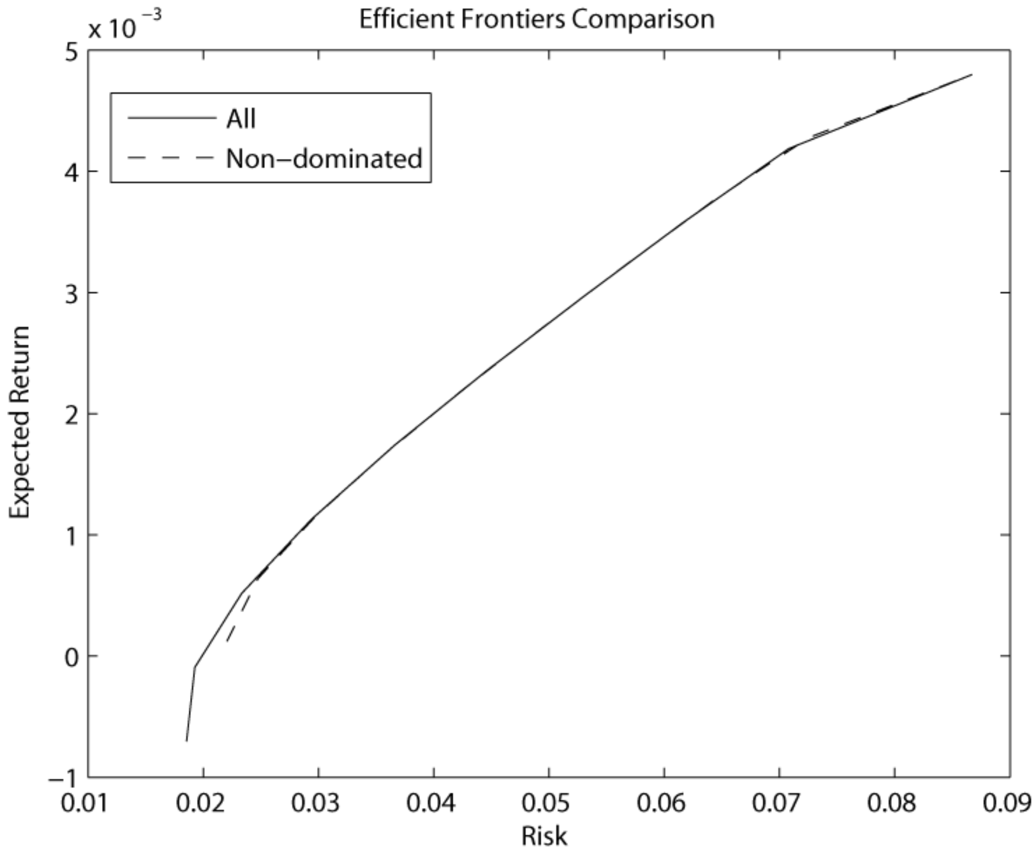

Figure 5.3: Comparison of the Pareto fronts. All-securities case and the non-dominated case are shown.

of gene i, the way to map these values into portfolio weights is

$$w_i = \frac{g_i}{\sum_{j=1}^{M} g_j}. \tag{5.9}$$

Equation 5.9 indicates each portfolio weight corresponds to the normalized value of g_i. This approach has the problem of the existence of redundant codes to describe the same portfolio. For example the portfolio $[0.5, 0.5]$ can be encoded by $[0001, 0001]$, $[0010, 0010]$, $[0011, 0011]$, etc. All of them are valid codes for the same portfolio. This has proved to be a nuisance for genetic algorithms because the optimal portfolio may have multiple genotypes, forcing the algorithm to consider each one of them as a distinct optimal solution.

Vector coding is proposed to reduce redundancy. Vector coding uses $M - 1$ genes to encode the portfolio. This approach visualizes portfolio weights as the components of an unity vector. It takes the advantage the sum of the squares of the components of an unity vector sums 1; the same occurs with portfolio's weights. In vector coding, genes encode the angles between the unitary vector and each coordinates axis plane. The weight w_i is defined in the following equation:

$$w_i = (c_i \sin(\theta_i))^2. \tag{5.10}$$

where θ_i is defined as

$$\theta_i = \left(\frac{\pi}{2}\right)\left(\frac{g_i}{2^b - 1}\right). \tag{5.11}$$

Equation 5.10 represents the i-th portfolio weight from $i = 1, 2, \ldots, M-1$. In equation 5.11, g_i is normalized using the maximum count possible with b bits and converted to radians to obtain θ_i. Figure 5.4 shows θ_i is the complement of the angle formed between the vector and the corresponding plane. The weights are computed from the component normal to the plane used to define θ_i. c_i is the magnitude of the projection of the vector on the plane used to define θ_i. In the beginning, $c_1 = 1$ because **C** is an unity vector. The next projection is computed from the following equation:

$$c_{i+1} = c_i \cos(\theta_i). \tag{5.12}$$

The last weight is the squared value of the last projection c_M:

$$w_M = (c_{M-1} \cos(\theta_{M-1}))^2. \tag{5.13}$$

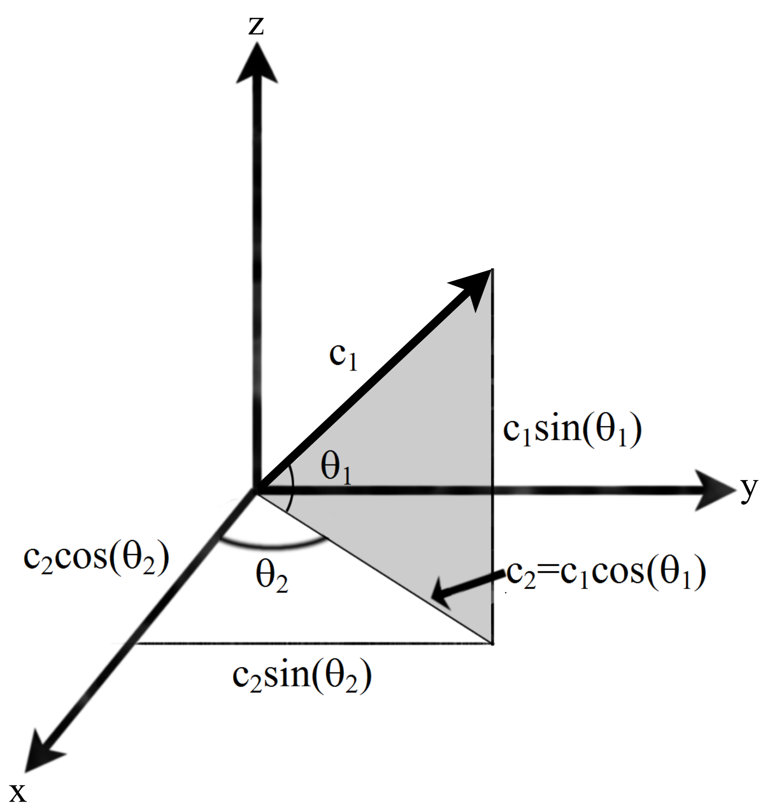

Figure 5.4: Graphical description of vector coding. The components of vector c are computed iteratively. The last component is the last projection c_{M-1}. The portfolio weights are obtained by squaring the component values.

5.2.2 Variable Duration Portfolio Sets

Algorithm 5.1.3 applies Sb-MORiOA iteratively to compute the multi-period Pareto front of portfolios. The dominated individuals are deleted until the Pareto front is complete. Nevertheless, a more efficient method was found to compute multi-period Pareto fronts. This modification allows to compute simultaneously different duration sets of portfolios.

Let us assume portfolios of M securities and T periods are to be codified using b bits. The naïve method explained above will need bMT bits to encode these portfolios. vector coding would require $b(M-1)T$ bits to encode the longest set. The simplest method to allow simultaneous optimization of different periods is including a gene to indicate the number of periods the set of portfolios should be applied. In that case, only a part of the total genes would be considered to describe the set of portfolios. Although, this approach would cause redundancy of solutions because the unused genetic material does not contribute to the fitness evaluation. Redundant individuals decrease the power of the algorithm to search solutions, because there is no way to distinguish the redundant individuals.

Although, this problem is bypassed when B&Hs portfolios are considered. B&Hs require bM to compute portfolio sets up to $2^b - 1$ periods. The virtual-gene genetic algorithm (vgGA) has capabilities to process genes of any base (Valenzuela-Rendón, 2003). This features can be applied to set the maximum duration to exactly T periods, avoiding redundancy. Only the first portfolio is encoded when using B&Hs, and the rest are obtained while evaluating equation 5.4. B&Hs have the advantage no transactions costs are incurred, except when a new portfolios set is chosen. Nevertheless, B&Hs require continuous revision to control risk. Portfolio revision usually incurs in transaction costs.

5.3 Methodology

The proposed method considers the information about the market's nature, the data history, the investor's preference, and the current state of the portfolio to make decisions. The market's nature information is considered in the model by the dynamic restrictions. The investor's preference is encoded through the goal and other parameters like the maximum number of periods of the initial Pareto front. The current state of the portfolio is evaluated using the data innovations.

The methodology to test how useful are these sources is the following: The Investment Strategies method was devised to include these information sources into the decision making process. Then, the improvement is measured through the performance of the method in real-world conditions. For this end, data from real markets is used to test it. The algorithm will be executed at random start points into the data. At those points, it will determine a goal according with the investor's preferences and the state of the market. Afterward, it will make decisions until the goal is attained or a maximum time limit is reached. The performance will be evaluated from the behavior of the algorithm against instances of different difficulty. In this context, difficulty refers to the probability the method has to reach the goal in time for the particular instance. Suitable performance metrics are discussed in later chapters.

Two benchmarks are proposed to draw conclusions about the usefulness of the considered information sources: Buy-and-holds and single-period portfolios. These methods represents two extremes about the use of information when making investment decisions. The buy-and-hold method takes information from one single set from the data history and makes no further changes. In this case, the method relies on market trends to make profit and avoid transaction costs. On the other hand, single-period portfolios apply the traditional portfolio optimization methods to make revisions at each period. Therefore, this method makes use of all the available information about the data history. The investment strategies method is at a middle point between this two approaches because it makes revisions like single-period portfolios, but also follow trends like buy-and-holds do. The comparison of the three methods should provide conclusions about how information affects the decision making process. Further detail is presented in later chapters.

5.4 Conclusions

This chapter showed the method to use Sb-MORiOA for financial decision making. The first part explained the equation which models the portfolio's wealth. This equation considers the effects of transaction costs, unbalance and the investor's expectations. A Monte-Carlo approach estimates the portfolio's wealth distribution from multiple realizations of the same individual. Sb-MORiOA is used to compute conditioned and unconditioned multi-period Pareto fronts. A method based on Investment Strategies was proposed to choose the next set of portfolios given data innovations. The method proposes a procedure to choose the most suitable securities.

The proposed method includes the investor's preference through the following parameters: the initial portfolio sets duration T, the maximum Pareto front risk σ_M, the minimum expected final portfolio value x_{Tmin}, the maximum risk expected $\alpha_{\sigma M}\sigma_M$, and the interest rate I. The proposed method provides investors evidence to help them determine their preferences. Two suggestions for I were made: I could be equal to inflation rate in the economy. Also, I could be equal to the lowest (or the average) return of the considered securities. Rules of thumb can be proposed to perform this step automatically. x_{Tmin} and $\alpha_{\sigma M}\sigma_M$ become the goal to guide the decision making process. The process can run until the goal is reached. Nevertheless, Investment Strategies can be applied while data innovations are available. Finally, the chapter proposed some custom modifications to implement portfolio selection with the Sb-MORiOA. Buy-and-hold portfolios coding and vector coding allow the algorithm to process different duration portfolios sets and reduce redundancy, respectively. The following chapters show experiments about the performance of the proposed method.

Chapter 6

Investment Strategies Evaluation

Investment strategies (ISs) were proposed in past chapters for automatic management of portfolios based on the investor's preferences and current market conditions. Their performance can be evaluated from the results obtained from the changes proposed by the algorithm to the portfolio. Several methods for portfolio evaluation can be found in the literature (Sharpe et al., 1999). Although, this task presents difficulties which have not been fully addressed in the traditional portfolio theory. This chapter explains some of the traditional methods. Besides, the limitations of these methods are discussed to formulate suggestions about the evaluation of investment strategies. The methods derived from the conclusions were implemented to evaluate the results of the experiments in later chapters.

The first part of this chapter explains some of the methods proposed in the literature. The second part makes an exposition about market indexes, which are widely accepted benchmarks for portfolio performance evaluation. The third part explains some of the limitations of this approach and proposes some alternatives to portfolio evaluation.

6.1 Portfolio Evaluation Methods

The portfolio evaluation methods can be classified in return measures and risk-weighted measures. Both approaches are explained below.

6.1.1 Return Measures

Sharpe et al. (1999) have discussed how both return and risk should be considered for portfolio evaluation. This reference has reported different manners to compute the portfolio's return. These methods are able to handle portfolios with cash flows, even when they occur within the period. Some of these methods are the dollar-weighted method and the time-weighted method.

The dollar-weighted return (or internal return) method consists on finding the value of the rate of return r in the following equation:

$$\sum_{i=0}^{N_p} \frac{C_i}{(1+r)^i} = C_0. \tag{6.1}$$

Equation 6.1 considers each cash flow C_i occurs sometime between the beginning and the end of the period. In this case, the frequency of the periods is adjusted to make cash flows to coincide with it. For example, if a single cash flow have occurred at the middle of the quarter, r is adjusted to be a semi-quarterly rate of return. N_p is the number of adjusted periods. The cash flows should be carefully examined to determine if they should be considered deposits or withdrawals from the portfolio.

The time-weighted return method considers the current portfolio value to measure the current rate of return. In this method, the rate of return from the beginning of the period to the moment of the cash flow is computed along with the rate of return computed from the cash flow to the end of the period. The final rate of return can be computed from the product of the partial rates of return. This procedure is repeated to obtain the return from any given period of time.

6.1.2 Risk-Adjusted Measures

The methods explained above are useful to measure the return of the portfolio when cash flows are present. Nevertheless, both methods overlook the risk of the decision. Sharpe et al. (1999) have discussed some methods to overcome this limitation. Most of these methods require a benchmark to make comparisons. Market indexes are commonly used for this end.

The traditional portfolio theory computes the return of the portfolio $\bar{r}_p$ and its risk σ_p with the following equations:

$$\bar{r}_p = \frac{1}{T}\sum_{i=1}^{T} r_p(i), \tag{6.2}$$

$$\sigma_p = \sqrt{\frac{1}{T-1}\sum_{i-1}^{T}(r_p(i) - \bar{r})}. \tag{6.3}$$

Equations 6.2 and 6.3 are simply the average and standard deviation of the portfolio's time-series of returns r_p. These values are used in the calculation of the risk-adjusted measures. Some of these methods are the ex-post β method, ex-post security market line, and the Sharpe's ratio. These methods are explained below.

Ex-Post-Characteristic Line Method

This method is based on the capital allocation pricing model (CAPM), which explains how individual securities are priced. The existence of the called *market portfolio* is assumed by the model. The behavior of the market portfolio is the best description of the market. Therefore, It will be the choice of all the rational investors independently from their individual preferences. In practice, a suitable benchmark is chosen instead of the market portfolio for evaluation purposes. Figure 6.1 illustrates both cases when the efficient frontier is composed by securities only and when it includes a fixed-income asset. The market portfolio is also indicated. The optimal decisions are combinations of the market portfolio and the fixed-income asset. That is the reason the efficient frontier which considers a fixed-income asset is a straight line.

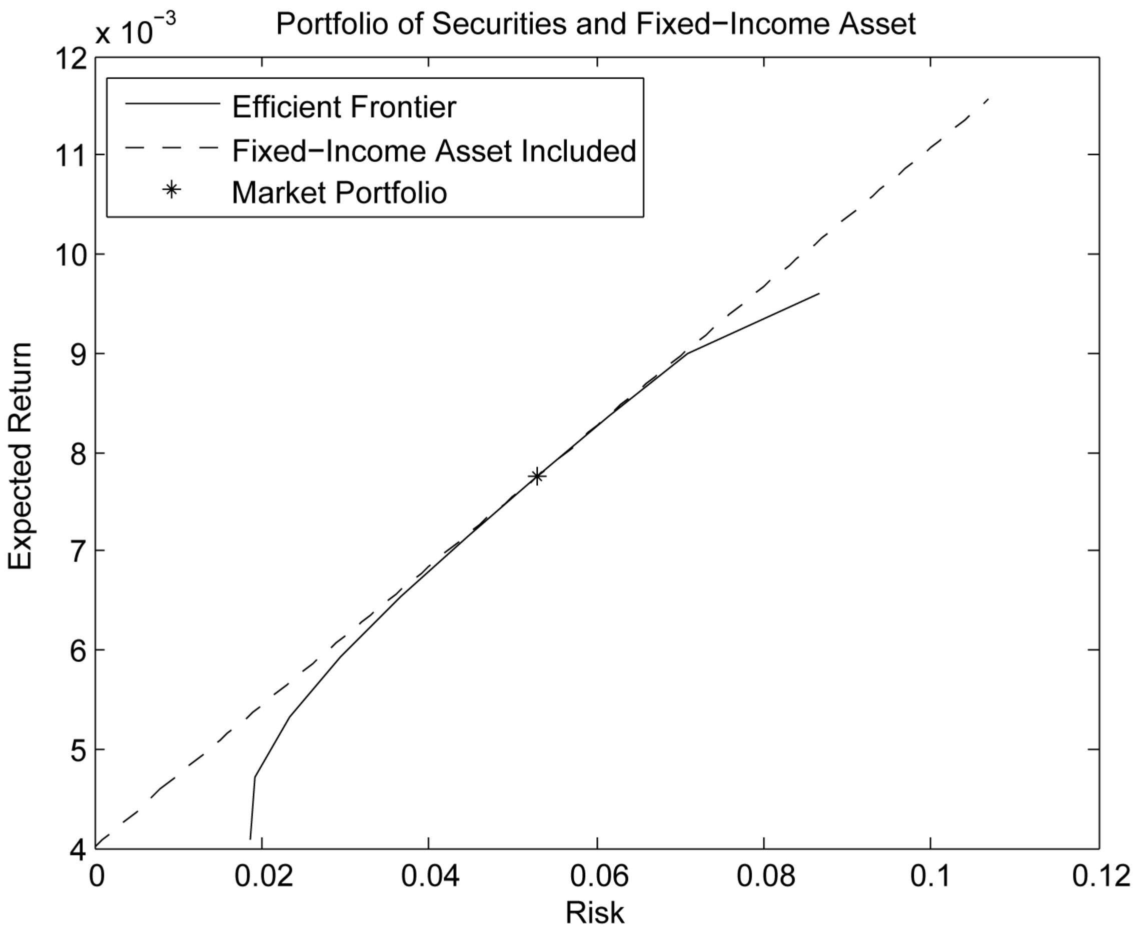

Figure 6.1: Examples of Efficient Frontiers

The return of these optimal portfolios are described by equation 6.4:

$$\bar{r}_p = r_f + \frac{\bar{r}_M - r_f}{\sigma_M}\sigma_p, \tag{6.4}$$

where $\bar{r}_p$ is the return of the portfolio and σ_p, its risk. $\bar{r}_M$ and σ_M refer to the market portfolio. r_f is the return of the fixed-income asset. Equation 6.4 is called the capital market line (CML). It can be modified to describe risk and return of individual securities. That is the case of the following equation:

$$\bar{r}_i = r_f + \frac{\bar{r}_M - r_f}{\sigma_M^2}\sigma_{iM}. \tag{6.5}$$

Equation 6.5 comes from the assumption each security's risk contributes to the total market portfolio's risk. Equation 6.5 can be also written written using the parameter β_{iM}:

$$\bar{r}_i = r_f + \left(\bar{r}_M - r_f\right)\beta_{iM}, \tag{6.6}$$

where β_{iM} is defined to be

$$\beta_{iM} = \frac{\sigma_{iM}}{\sigma_M^2}. \tag{6.7}$$

Equation 6.6 is called the security market line (SML). In equation 6.7, σ_{iM} is the covariance between the security and the market portfolio. σ_M^2 is the variance of the market portfolio. The ex-post (i.e. after the fact) term refers the value of β is computed from historical data. Excess returns are used for the sake of this calculation. Excess returns are defined to be

$$r_{ep} = r_p - r_f. \tag{6.8}$$

Equation 6.8 means excess returns are the ones exceeding the return of the fixed-income asset. The covariance and the variance from equation 6.7 are computed using excess returns.

The values of β_{iM} can be used to compare the risk of different stocks, but equation 6.6 can be modified to compare portfolios instead of individual assets. The resulting equations is

$$\alpha_p = \bar{r}_p - \left(r_f + \left(\bar{r}_{bp} - r_f\right)\beta_p\right). \tag{6.9}$$

Equation 6.9 computes the difference of the returns between the portfolio and the benchmark, which is denoted α_p. The firs term is the actual average return of the portfolio. The second term is the evaluation of the portfolio return using equation 6.6. β_p is the ratio of the covariance between the portfolio and the benchmark and the benchmark's variance. Positive values of α_p indicate the evaluated portfolio is better than the benchmark.

Sharpe's Ratio

This measure is derived from the capital market line model (equation 6.4). The Sharpe's ratio Sh_p for the portfolio is defined as

$$\text{Sh}_p = \frac{\bar{r}_p - r_f}{\sigma_p}. \tag{6.10}$$

The value of Sh_p is compared against the slope of equation 6.4 to determine if the portfolio is better than the benchmark. The difference between this method and the ex-post characteristic line method is Sh_p is based on the total risk instead of the market risk only. Total risk can be separated in

$$\sigma_i^2 = \beta_{iM}^2 \sigma_M^2 + \sigma_{\epsilon i}^2. \tag{6.11}$$

Equation 6.11 indicates β_{iM} is only a fraction of the risk of a portfolio. The risk not explained by the model is considered random noise.

6.2 Market Indexes

The explained methods are based on the capital allocation market model. The existence of the market portfolio is one of the strongest assumptions of this model. A relevant benchmark is used instead to apply these methods for portfolio performance purposes. Market indexes are usually chosen for this end.

Indexes are instruments used to measure the state of the market. A specific sector can be traced instead, depending on the securities conforming the index. The Dow Jones industrial average (DJI), the Standard&Poor 500 (S&P500), The Índice de Precios and Cotizaciones (IPC), the Nikkei 225, or the FTSE 100 are examples of marked indexes. A brief description of some market indexes appears in table 6.1. The information was taken from the following references: (*Barron's 400*, 2014, November 1st; *S&P Dow Jones Indices*, 2014, November 1st; *FTSE*, 2014, December; *Metodología del IPC*, 2014, September 1st; *NASDAQ Global Indexes*, 2015, February; *Nikkei Indexes*, 2015, February 2nd; *S&P Dow Jones Indices*, 2014, November 1st).

The composition of market indexes changes with time because their components should be a relevant sample of the economy. Besides, some indexes impose special requirements to their component stocks. For example, a component from the IPC should have a at least 12% of the shares outstanding to be free-floating to be considered a part of the index (*Metodología del IPC*, 2014, September 1st).

There are different methods to compute a market index. A brief introduction to some of them is presented. The price weighting method, the capitalization weighting method, and the equal weighting method are explained below.

Price Weighting Method (PW)

This type of market index is computed from the weighted sum of its components:

$$P(t) = \frac{\sum_{i=1}^{M} p_i(t)}{d(t)}. \tag{6.12}$$

Where $p_i(t)$ is the price of the i-th asset and $P(t)$ is the index value. The average is the simplest case of equation 6.12. Although, The value of the divisor $d(t)$ is not restricted to M only. On the contrary, $d(t)$ is constantly updated to compensate the effect of changes in the index composition and splits. Splits is the division of the value of securities. For example, a security of value $\$10$ will be equivalent to 2 securities of values $\$5$ after the split. Companies split their securities to control their prices and liquidity. $d(t)$ is adjusted by solving the following equation:

$$\frac{\sum_{i=1}^{M} p_i(t)}{d(t)} = \frac{\sum_{i=1}^{M} p_i(t) s_i(t)}{d(t-1)}. \tag{6.13}$$

Where $s_i(t)$ represents the split value of the i-th asset. $s_i(t) = 1$ when no split was applied to the security. A similar procedure is applied when securities are substituted by others into the index composition.

Capitalization Weighting Method (CW)

Capitalization weighted indexes are computed in the following manner:

$$P(t) = P(0) \left(\frac{\sum_{i=1}^{M} p_i(t) o_i(t)}{\sum_{i=1}^{M} p_i(0) o_i(0)} \right). \tag{6.14}$$

Equation 6.14 indicates securities prices are multiplied by the number of securities outstanding at time t. This is represented by $o_i(t)$. This product is divided by the index value at time 0. $P_i(0)$ and time 0 are chosen arbitrarily. This method has the advantage large companies are more heavily weighted than small ones. Besides, it does not require further considerations about splits because they are included in the number of shares outstanding for a particular security.

Equal Weighting Method (EW)

This method computes the value of the index in the following way:

$$P(t) = \frac{P(t-1)}{M} \sum_{i=1}^{M} \frac{p_i(t)s_i(t)}{p_i(t-1)} \tag{6.15}$$

Equation 6.15 computes the average of the price relatives (i.e. the ratio of present and past prices) to compute the new value of the index. $s_i(t)$ is the split value of the i-th asset, and M is the number of components of the index. The geometric mean could be also used to compute the index value.

Table 6.1: Examples of Market Indexes

Market Index	Components	Market	PW	CW	EW
Barron	400	US Public Companies			√
DJI	30	NYSE, NASDAQ	√		
FTSE	100	London SE		√	
IPC	35	Bolsa Mexicana de Valores		√	
NASDAQ	100	NASDAQ		√	
Nikkei	225	Tokyo SE	√		
S&P	500	NYSE, NASDAQ		√	

6.3 Limitations of Portfolio Evaluation Methods

Some methods to evaluate portfolios has been presented in the past sections. Their main limitation is the distribution of final return is computed from the time-series of the portfolio's returns, as shown in equations 6.2 and 6.3. This means the final expected return and the volatility of the final return are estimated from the time-series of the portfolio. This computation assumes the standard deviation of the time series is a good estimation of the volatility of the final return. The volatility of the market indexes is also estimated in a similar manner. This procedures is proposed in the traditional portfolio theory because the estimation of volatility is a difficult task. General auto-regressive conditional heteroskedasticity (GARCH) models are an example of stochastic time-series models which are used for this end.

Nevertheless, the estimation of the volatility of the distribution of final return can be estimated using a Monte-Carlo approach. Multiple realizations can be generated to estimate the distribution of the final return and use them to evaluate the portfolios. In this work, this approach is now proposed to compute the risk and the final return of the portfolios instead of using equations 6.2 and 6.3. The return and risk obtained by this method can be applied to any of the performance measures explained above.

In past sections it was explained that market indexes are frequently considered benchmarks to measure the performance of portfolios. Measures based on the capital allocation market model (CAPM) rely heavily on market indexes to substitute the ideal market portfolio considered in the model. Nevertheless, market indexes are not portfolios; this means they are not subject to dynamic restrictions like transaction costs, unbalance, or inflation. For example, the Dow Jones index value is obtained with the price-weighted method, and is equivalent to an equally weighted portfolio. Nevertheless, unbalance and transaction costs are ignored when computing the next index value. Besides, its value can be higher than the sum of individuals stocks depending on the current value of the divisor. Both situations are impossible with regular portfolios. Similar situations are present in capitalization-weighted indexes. In this work, some benchmarks with closer behavior to portfolios are preferred instead of market indexes. The experiments considered comparisons against single-period portfolios computed with Markowitz's method. Buy-and-holds are also proposed for comparison purposes.

Moreover, the Sharpe's ratio was the risk-adjusted measure preferred because its calculation does not depended directly of the performance of a market index. The computation of the risk and return of a market index becomes difficult when equation 6.2 and 6.3 are not used. This occurs because the required information to obtain these values at any time needed is not readily available. For example, information about the divisor used to compute the Dow Jones index is reported only in periodic (not daily) reports. In another example, the current number of outstanding stocks is needed to compute capitalization-weighted index values, but it is also reported at certain periods only. The Sharpe's ratio provides a risk-weighted measure of portfolio performance free of these problems.

6.4 Experiments and Data Set

Experiments are presented in the following chapter. Data from the Dow Jones index (DJI) and the Mexican Índice de Precios y Cotizaciones (IPC) were chosen for the experiments due their relevance to domestic economy. The adjusted close data are used for the tests because they include the effects of splits and dividends. The data are openly available online (*Yahoo! Finance*, 2015, February). Although, the recollection of data form long past dates became increasingly difficult. The reasons for the scarcity of data are diverse:

- Companies from the past could have been bankrupted and disappeared. Data about a bankrupted company becomes scarce because there is no one to be responsible to keep and maintain it. For example, General Motors Co. (GM) was part of DJI until September 22th, 2008, when it was declared bankrupt (*Yahoo! Finance*, 2015, February). Data from their bleak times were retired from the GM ticker. A new company was created to handle GM liquidation. Data generated after the bankrupt is available under the MTLQQ ticker only (*InvestorPoint.com. Investor Information Systems*, 2015, February).

- Companies could have changed names or merge with others. Data can become confusing at certain times. They should be changed accordingly with these events. For example, Southern Bell Communications (SBC) was part of DJI until November 21th, 2005. SBC merged with AT&T Wireless and kept the AT&T ticker. SBC ticker is no longer available, but the information of historical prices can be found at the AT&T site (*AT&T Historical Quote*, 2015, February).

- Old data can be simply anymore available. Bolsa Mexicana de Valores (BMV) is responsible to compute the IPC. IPC data are available from November 8th, 1991 to the date. Although, information about the index weights is available only from June 1st, 2012 (*Notas Sobre Índices*, 2015, February) to the date only. This fact limits the available data for the experiments.

The composition of DJI appears in table 6.4. History was collected from November 1st, 1999 to November 3rd, 2014. The limitations explained above prevented the recollection of further data. The company names are shown in table 6.2. The reported dates correspond with changes of the index. A similar description for the IPC appears in tables 6.5 and 6.6. The tickers description is shown in tables 6.3. The IPC sample is smaller because the information about this index is more limited. Nevertheless, more companies changes occurred at the IPC during the sample time. This is because the tight restrictions about the minimum shares outstanding requirement (*Metodología del IPC*, 2014, September 1st).

6.5 Conclusion

This chapter presented some common portfolio evaluation methods used by financial professionals. The methods attempt to consider both risk and return when comparing portfolios. Most of these methods rely on benchmarks, and market indexes are commonly used for this end.

This chapter presented some limitations about the traditional evaluation methods. The main limitations are the difficulty to compute the distribution of the final return and volatility. A Monte-Carlo method was proposed to avoid this difficulty. Moreover, the use of market indexes overlooks dynamic restrictions like transaction costs and unbalance which cannot be ignored in real-world situations. Buy-and-holds and single-period portfolios were proposed as benchmarks for comparison instead of marked indexes because they are subject to dynamic restrictions even when their optimization process ignores them. Nevertheless, securities from DJI and IPC were proposed as the data sets to be used in the experiments. The Sharpe's ratio is computed in the experiments to have a risk-weighted measure of the portfolio's distribution of returns.

Finally, the chapter presented information about the data sets to be used in the experiments. The DJI and the IPC were chosen due their impact in the domestic Mexican economy. The last sections explained the limitations about the ability to collect data from the past. Tables were used to present the content of the indexes and their changes along the years.

Table 6.2: Tickers from DJI

Ticker	Company
AA	Alcoa Inc.
AIG	American International Group, Inc.
AXP	American Express Company
BA	The Boeing Company
BAC	Bank of America Corporation
C	Citigroup Inc.
CAT	Caterpillar Inc.
CSCO	Cisco Systems, Inc.
CVX	Chevron Corporation
DD	E. I. du Pont de Nemours and Company
GE	General Electric Company
GM	General Motors Company
GS	The Goldman Sachs Group, Inc.
HD	The Home Depot, Inc.
HON	Honeywell International Inc.
HPQ	Hewlett-Packard Company
IBM	International Business Machines Corporation
IP	International Paper Company
JNJ	Johnson & Johnson (JNJ)
KO	The Coca-Cola Company
KODK	Eastman Kodak Co.
KRFT	Kraft Foods Group, Inc.
MCD	McDonald's Corp.
MMM	3M Company
MO	Altria Group Inc.
MRK	Merck & Co. Inc.
MSFT	Microsoft Corporation
NKE	Nike, Inc.
PFE	Pfizer Inc.
PG	The Procter & Gamble Company
T	AT&T, Inc.
TRV	The Travelers Companies, Inc.
UNH	UnitedHealth Group Incorporated
UTX	United Technologies Corporation
V	Visa Inc.
VZ	Verizon Communications Inc.
WMT	Wal-Mart Stores Inc.
XOM	Exxon Mobil Corporation

Table 6.3: Tickers from IPC

Ticker	Company
ALFAA	Alfa
ALPEK-A	Alpek
ALSEA	Alsea
AMXL	America Movil
ARA	Consorcio ARA
ASURB	Grupo Aeroportuario del Sureste
AXTELCPO	Axtel
AZTECACPO	TV Azteca
BIMBOA	Grupo Bimbo
BOLSAA	Bolsa Mexicana de Valores
CEMEX-CPO	CEMEX
CHDRAUIB	Grupo Comercial Chedraui
COMERCIUBC	Controladora Comercial Mexicana
COMPARC	GENTERA
ELEKTRA	Grupo Elektra
FEMSAUBD	Fomento Econ´ómico Mexicano
GAPB	Grupo Aeroportuario del Pacífico
GCARSOA1	Grupo Carso
GENTERA	GENTERA
GEOB	Corporación GEO
GFINBURO	Grupo Financiero Inbursa
GFNORTEO	Grupo Financiero Banorte
GFREGIOO	BANREGIO Grupo Financiero
GMODELOC	Grupo Modelo
GSANBOR	Grupo Sanborns
HOMEX	Desarrolladora Homex
ICA	Empresas ICA
ICHB	Industrias CH
IENOVA	IENOVA
KIMBERA	Kimberly - Clark de México
KOF	Coca-Cola FEMSA
LABB	Genomma Lab Internacional
LALA-B	Grupo LALA
LIVEPOL1	El Puerto de Liverpool
MEXCHEM	Mexichem
MEXICOB	Grupo M´éxico
MFRISCO-A-1	Mineras FRISCO
OHLMEX	OHL México
PE&OLES	Industrias Penoles
PINFRA	Promotora y Operadora de Infraestructura
SANMEXB	Grupo Financiero Santander México
SORIANAB	Organizacion Soriana
TLEVISACPO	Grupo Televisa
URBI	Urbi, Desarrollos Urbanos
WALMEXV	Wal-Mart de México

Table 6.4: Composition of DJI

Date	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
9/23/2013	AXP	BA	CAT	CSCO	CVX	DD	DIS	GE	GS	HD
9/24/2012	AXP	BA	CAT	CSCO	CVX	DD	DIS	GE	AA	HD
6/8/2009	AXP	BA	CAT	CSCO	CVX	DD	DIS	GE	AA	HD
9/22/2008	AXP	BA	CAT	C	CVX	DD	DIS	GE	AA	HD
2/19/2008	AXP	BA	CAT	C	CVX	DD	DIS	GE	AA	HD
11/21/2005	AXP	BA	CAT	C	HON	DD	DIS	GE	AA	HD
4/8/2004	AXP	BA	CAT	C	HON	DD	DIS	GE	AA	HD
1/27/2003	AXP	BA	CAT	C	HON	DD	DIS	GE	AA	HD
11/1/1999	AXP	BA	CAT	C	HON	DD	DIS	GE	AA	HD
Date	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
9/23/2013	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	NKE
9/24/2012	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	BAC
6/8/2009	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	BAC
9/22/2008	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	BAC
2/19/2008	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	BAC
11/21/2005	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	MO
4/8/2004	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	MO
1/27/2003	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	MO
11/1/1999	IBM	INTC	JNJ	JMP	KO	MCD	MMM	MRK	MSFT	MO
Date	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30
9/23/2013	PFE	PG	T	TRV	UNH	UTX	V	VZ	WMT	XOM
9/24/2012	PFE	PG	T	TRV	UNH	UTX	HPQ	VZ	WMT	XOM
6/8/2009	PFE	PG	T	TRV	KRFT	UTX	HPQ	VZ	WMT	XOM
9/22/2008	PFE	PG	T	GM	KRFT	UTX	HPQ	VZ	WMT	XOM
2/19/2008	PFE	PG	T	GM	AIG	UTX	HPQ	VZ	WMT	XOM
11/21/2005	PFE	PG	T	GM	AIG	UTX	HPQ	VZ	WMT	XOM
4/8/2004	PFE	PG	SBC	GM	AIG	UTX	HPQ	VZ	WMT	XOM
1/27/2003	KODK	PG	SBC	GM	T	UTX	HPQ	IP	WMT	XOM
11/1/1999	KODK	PG	SBC	GM	T	UTX	HPQ	IP	WMT	XOM

Table 6.5: Composition of IPC

Date	S1	S2	S3	S4	S5	S6	S7
9/1/2014	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
6/2/2014	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
3/3/2014	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
12/2/2013	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
6/3/2013	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
3/1/2013	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
12/3/2012	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
9/3/2012	AC	ALFA	ALPEK	ALSEA	AMX	ASUR	BIMBO
6/1/2012	AC	ALFA	ARA	ALSEA	AMX	ASUR	BIMBO
Date	S8	S9	S10	S11	S12	S13	S14
9/1/2014	BOLSA	CEMEX	COMERCI	ELEKTRA	FEMSA	GAP	GCARSO
6/2/2014	BOLSA	CEMEX	COMERCI	ELEKTRA	FEMSA	GAP	CHDRAUI
3/3/2014	BOLSA	CEMEX	COMERCI	ELEKTRA	FEMSA	GAP	CHDRAUI
12/2/2013	BOLSA	CEMEX	COMERCI	ELEKTRA	FEMSA	GAP	CHDRAUI
6/3/2013	BOLSA	CEMEX	AZTECA	ELEKTRA	FEMSA	GAP	CHDRAUI
3/1/2013	BOLSA	CEMEX	AZTECA	ELEKTRA	FEMSA	GAP	CHDRAUI
12/3/2012	BOLSA	CEMEX	AZTECA	ELEKTRA	FEMSA	GAP	CHDRAUI
9/3/2012	BOLSA	CEMEX	AZTECA	ELEKTRA	FEMSA	GAP	CHDRAUI
6/1/2012	BOLSA	CEMEX	AZTECA	ELEKTRA	FEMSA	GAP	CHDRAUI
Date	S15	S16	S17	S18	S19	S20	S21
9/1/2014	GENTERA	GFINBUR	GFNORTE	GFREGIO	GMEXICO	GRUMA	ICA
6/2/2014	GENTERA	GFINBUR	GFNORTE	GFREGIO	GMEXICO	GRUMA	ICA
3/3/2014	GENTERA	GFINBUR	GFNORTE	GFREGIO	GMEXICO	GRUMA	ICA
12/2/2013	COMPARC	GFINBUR	GFNORTE	GFREGIO	GMEXICO	GRUMA	ICA
6/3/2013	COMPARC	GFINBUR	GFNORTE	GEO	GMEXICO	GRUMA	ICA
3/1/2013	COMPARC	GFINBUR	GFNORTE	GEO	GMEXICO	GRUMA	ICA
12/3/2012	COMPARC	GFINBUR	GFNORTE	GEO	GMEXICO	GRUMA	ICA
9/3/2012	COMPARC	GFINBUR	GFNORTE	GEO	GMEXICO	GRUMA	ICA
6/1/2012	COMPARC	AXTEL	GFNORTE	GEO	GMEXICO	GRUMA	ICA

Table 6.6: Composition of IPC (Cont.)

Date	S22	S23	S24	S25	S26	S27	S28
9/1/2014	ICH	IENOVA	KIMBER	KOF	LAB	LALA	LIVEPOL
6/2/2014	ICH	IENOVA	KIMBER	KOF	LAB	GSANBOR	LIVEPOL
3/3/2014	ICH	IENOVA	KIMBER	KOF	LAB	GSANBOR	LIVEPOL
12/2/2013	ICH	IENOVA	KIMBER	KOF	LAB	GSANBOR	LIVEPOL
6/3/2013	ICH	HOMEX	KIMBER	KOF	LAB	MFRISCO	LIVEPOL
3/1/2013	ICH	HOMEX	KIMBER	KOF	LAB	MFRISCO	LIVEPOL
12/3/2012	ICH	HOMEX	KIMBER	KOF	LAB	MFRISCO	LIVEPOL
9/3/2012	ICH	HOMEX	KIMBER	KOF	LAB	MFRISCO	LIVEPOL
6/1/2012	COMERCI	HOMEX	KIMBER	SORIANA	LAB	MFRISCO	LIVEPOL
Date	S29	S30	S31	S32	S33	S34	S35
9/1/2014	MEXCHEM	OHLMEX	PE&OLES	PINFRA	SANMEX	TLEVISA	WALMEX
6/2/2014	MEXCHEM	OHLMEX	PE&OLES	PINFRA	SANMEX	TLEVISA	WALMEX
3/3/2014	MEXCHEM	OHLMEX	PE&OLES	PINFRA	SANMEX	TLEVISA	WALMEX
12/2/2013	MEXCHEM	OHLMEX	PE&OLES	PINFRA	SANMEX	TLEVISA	WALMEX
6/3/2013	MEXCHEM	OHLMEX	PE&OLES	URBI	SANMEX	TLEVISA	WALMEX
3/1/2013	MEXCHEM	OHLMEX	PE&OLES	URBI	GMODELO	TLEVISA	WALMEX
12/3/2012	MEXCHEM	OHLMEX	PE&OLES	URBI	GMODELO	TLEVISA	WALMEX
9/3/2012	MEXCHEM	OHLMEX	PE&OLES	URBI	GMODELO	TLEVISA	WALMEX
6/1/2012	MEXCHEM	OHLMEX	PE&OLES	URBI	GMODELO	TLEVISA	WALMEX

Chapter 7

Experiments Results

Past chapters have explained Sb-MOEA and its modifications to perform risk optimization (Sb-MORiOA). Also, the investment strategies (ISs) method was proposed to manage portfolios considering dynamic restrictions and the investor's preference. This chapter presents the results of the experiments conducted to evaluate the proposed method. The investment strategies method was implemented using Sb-MORiOA.

The first part of this chapter describes the experiments and the evaluation parameters. The second part shows explain the results. The last section shows the discussion and the obtained conclusions.

7.1 Experiments Description

Data from the Dow Jones industrial average (DJI) and the Índice de Precios y Cotizaciones (IPC) were selected for the experiments. A discussion about these data sets was presented in past chapters. A random starting date is selected for each experiment. From that point, the algorithm managed the portfolio to reach the goal, which is determined with the procedure explained in chapter 5. The goal was set from the first Pareto front, which was computed with a fixed number of periods $T = 5$ and $\alpha_{\sigma M} = 0.8$. The interest rate I was computed to be the average return of the selected securities by the algorithm for the particular experiment. The possibility to use a fixed-income asset was included. Cash was used for this end. The transaction cost rate was set to be $c = 0.005$. A maximum number of periods $T_{\max}$ were given to the algorithm to reach the goal. Time windows ensure all the experiments had an equal opportunity to perform the task. Besides, a sliding window method is used to constantly update the history. Used data became part of data history and the oldest ones were discarded. The probability distribution of stock returns was updated using the current history when the algorithm required it. Starting points unable to comply to these restrictions were not considered for the experiments. DJI experiments used $T_{\max} = 120$ and IPC experiments used $T_{\max} = 60$. These values were empirically selected to make time duration of experiments reasonable. 30 random instances were chosen from each data set for the experiments.

The experiments tested five strategies with different levels of risk: $S_0 = [0, 1]$, $S_{25} = [0.25, 0.75]$, $S_{50} = [0.5, 0.5]$, $S_{75} = [0.75, 0.25]$, and $S_1 = [1, 0]$. Investment strategies were

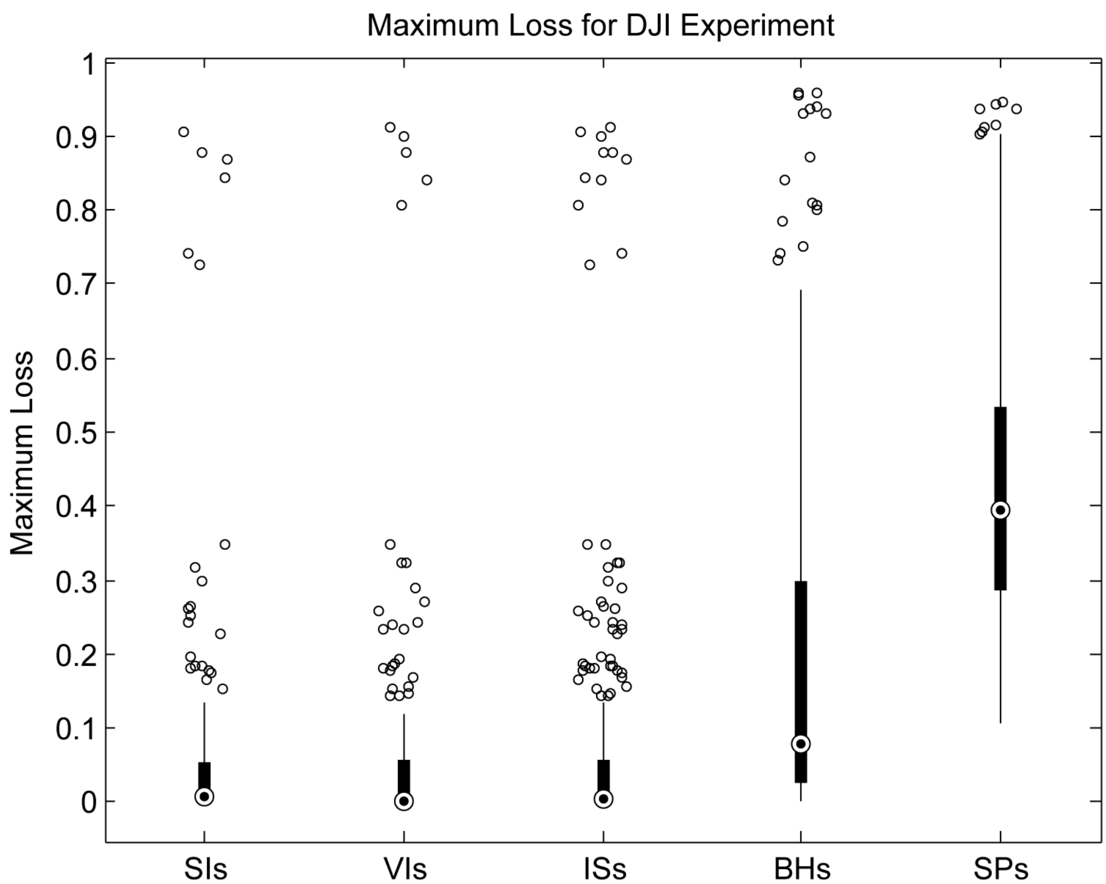

Figure 7.1: Average Maximum Loss for DJI Experiment per Method

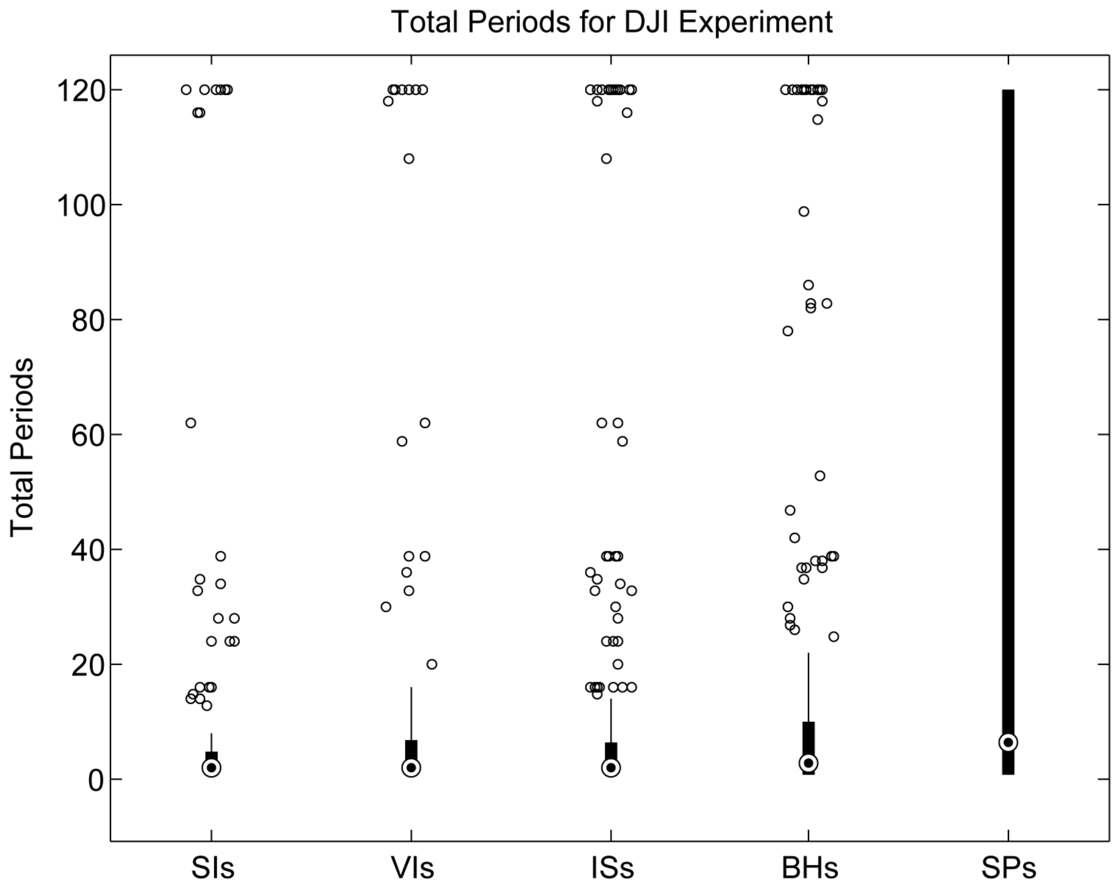

Figure 7.2: Total Periods for DJI Experiment per Method

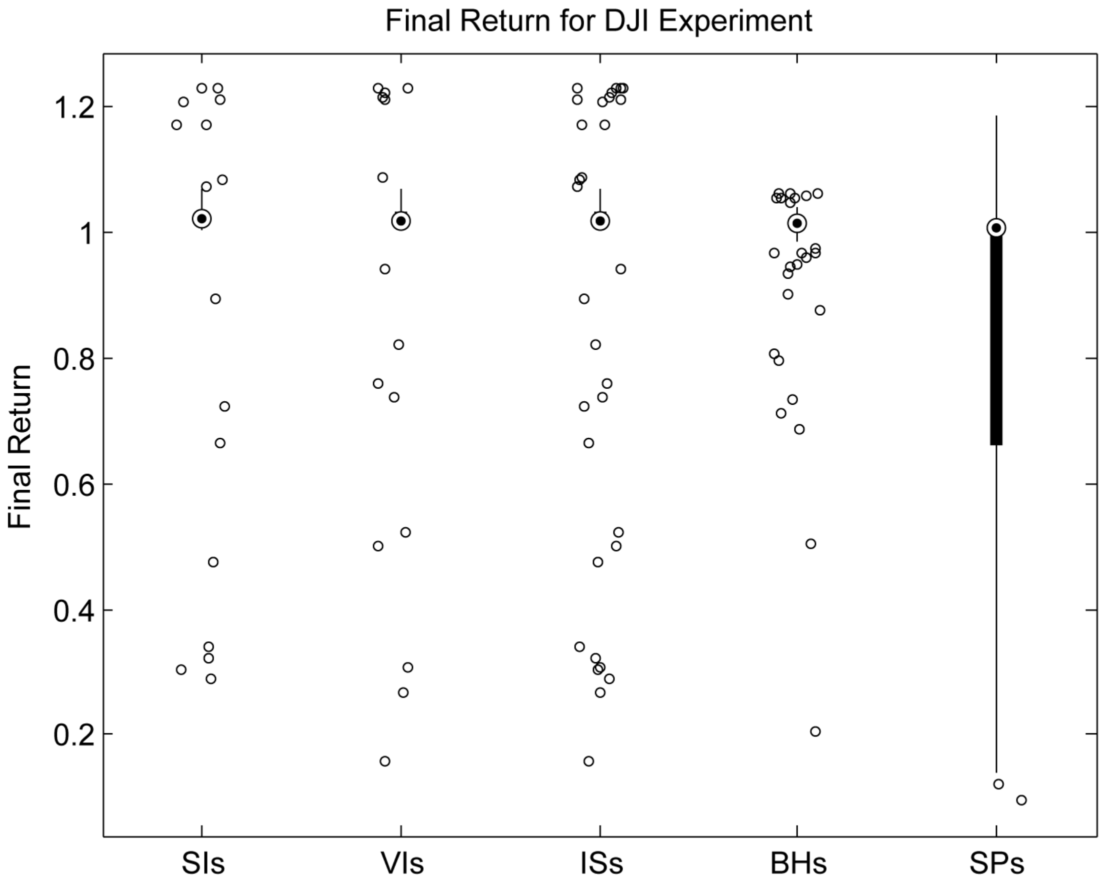

Figure 7.3: Final Return for DJI Experiment per Method

used to select the next set of portfolios to be used by the algorithm. Besides, the experiments tested two different risk measures: The standard deviation of the final portfolio value and the value-at-risk (VaR) of the final portfolio value. Value-at-risk can be understood as the α-quantile of the distribution. Both risk measures were applied to the same data sets to investigate their effect in performance.

The number of individuals in the population and the number of cycles are determined by equation 7.1:

$$n_{\text{inds}} = c_1 \lceil \log(T) \rceil . \quad (7.1)$$

Where $c_1 = 50$ for the case of individuals, and $c_1 = 100$ for the case of cycles.

The crossover probability was set to $p_c = 1$, and the mutation probability was $p_m = 0.1$. The maximum number of evaluations per individual was 10000. The number of bits per portfolio weight was $b = 4$. The total number of bits depended on the number of securities selected by the algorithm, which was determined using the procedure described in chapter 5.

Two other investment methods were tested for comparison purposes. The first one is buy-and-hold (B&H), which is widely used by investors. The same securities chosen by the

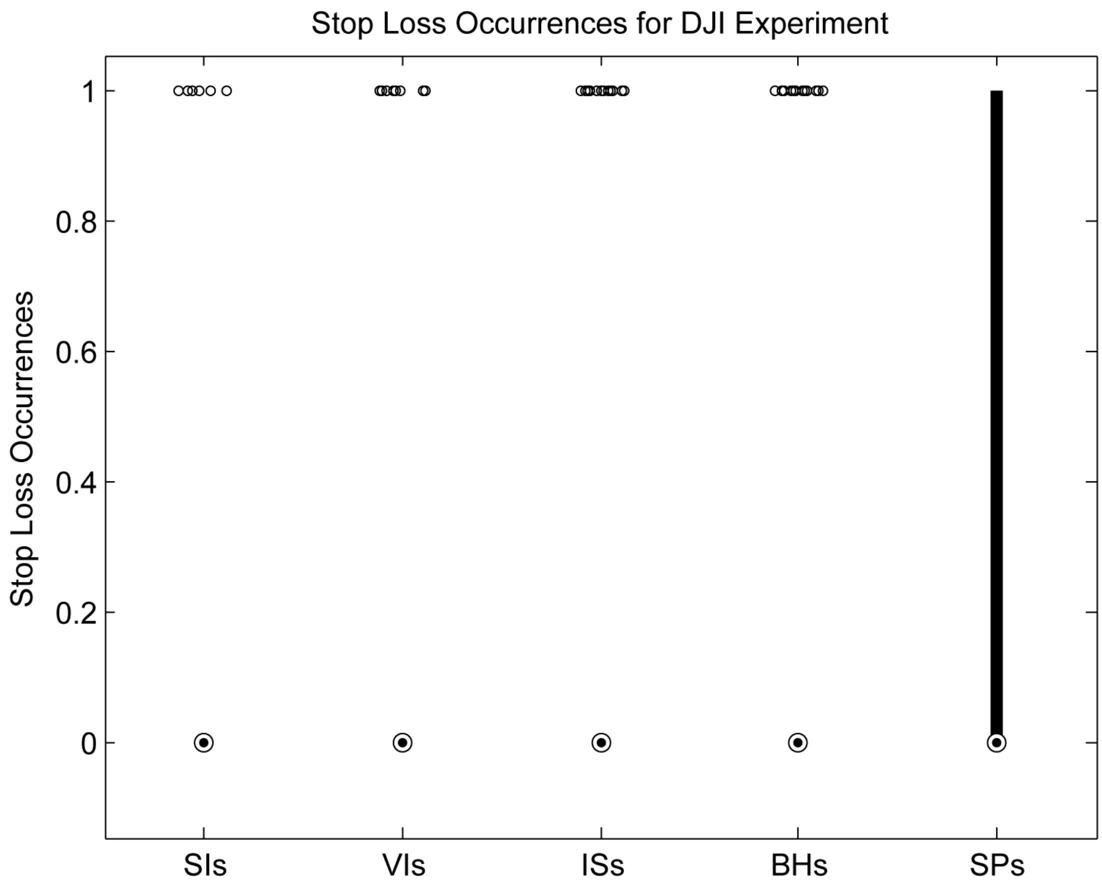

Figure 7.4: Stop Loss for DJI Experiment per Method

algorithm were used for the B&Hs. They are determined from the Markowitz's efficient frontier, and 10 portfolios were chosen from it. The selected portfolios were hold until the goal was reached or time was over. These portfolios were not re-balanced during the experiment. In the results, $S1$ denotes the lowest risk portfolio. The second method is a single-period investment method where the optimal portfolio was computed at each period with the Markowitz's method. 10 portfolios were selected from the efficient frontier for the test, where $M1$ denotes the lowest risk portfolio. These portfolios were tested against the same random instances used to evaluate the investment strategies.

7.1.1 Measures of Performance

The same sets of random instances were used to test the different methods. This allowed a direct comparison of the different methods for the same conditions. The overall performance can be computed from the results obtained by a particular method from the different instances. The performance measures considered were the following: Maximum loss, total periods, final return, stop loss time, expected return, risk, Sharpe's ratio, and the average periods per decision. Maximum loss refers to the minimum portfolio value reached during the experiment.

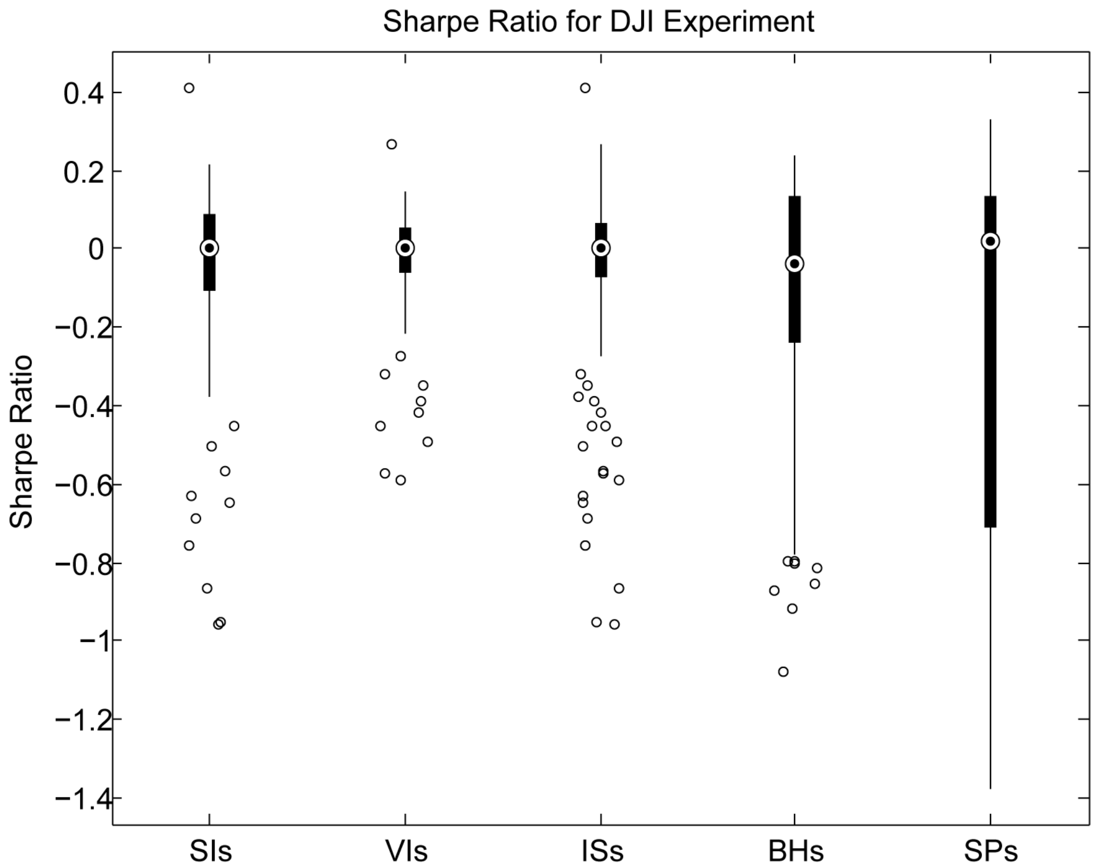

Figure 7.5: Sharpe's Ratio for DJI Experiment per Method

Total periods is the time used by the algorithm to reach the goal. The total periods was equal to $T_{\max}$ when the algorithm was unable to reach the goal in time. The final return is usually close to the goal because the algorithm stops when the goal is reached. Nevertheless, this measures is useful to analyze the cases when the algorithm failed the task. Stop loss time refers to the period when the portfolio wealth dropped below a certain value. In the experiments, a stop loss occurrence is counted when portfolio value drops below 70% of the initial value. Investors usually apply stop loss to prevent catastrophe. The experiment did not stopped when a stop loss had place, but it was registered to investigate the benefit of this practice.

Equations 6.2 and 6.3 indicate the current theory estimates risk and final return of portfolios based on the time-series of the current portfolio value. Nevertheless, the Monte-Carlo approach allows the estimation of the volatility and the final return directly. The Sharpe's ratio is used to investigate the performance of investment strategies, buy-and-holds and single-period policies, but the estimation of risk and return were computed using the Monte-Carlo approach instead of the traditional method. The risk and returns of the portfolios were plotted at the risk-return plane to investigate their domination relationships.

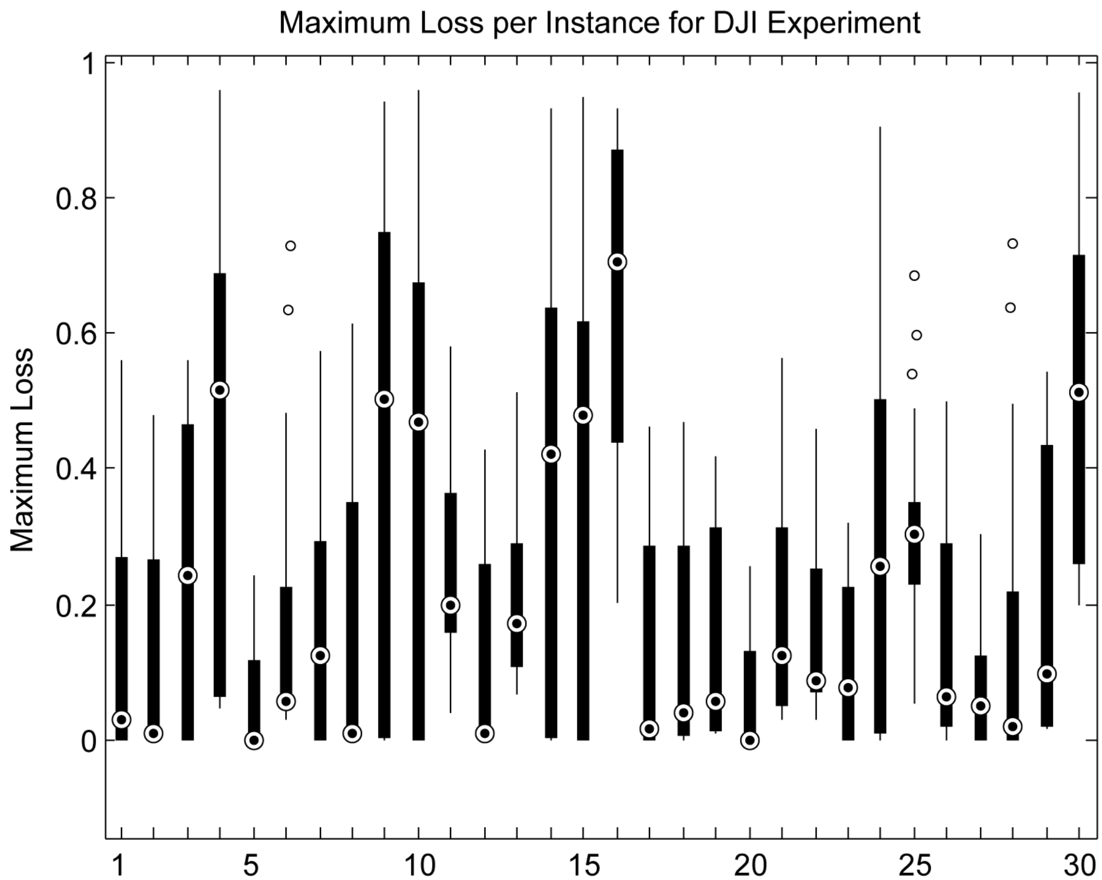

Figure 7.6: Average Maximum Loss for DJI Experiment per Instance

7.1.2 Box Plots Method

Box plots were chosen to present the results from the experiments. This method was preferred because it provides further information about the distribution of data. For example, box plots identify outliers clearly, which could mislead results if they are not considered when interpreting data. The figures below show outliers were a common occurrence in the results, therefore this method is useful in this case. The box represents the area from first quartile to the third quartile of the distribution. The dot inside the box represents the median, which comparison has statistical significance. The length of the whiskers indicates the minimum and maximum data which are not considered outliers. The maximum whisker extension is $3/2$ of the size of the box. Data beyond those limits are considered outliers.

Hypothesis testing was conducted to investigate the significance of the conclusions obtained by the box plot method. The Kruskal-Wallis test is usually applied to compare the medians of different samples (McDonald, 2015). It is a non-parametric test, this means it does not assumes a particular distribution for the data. In general, the test explores if the ranked samples come from the same distribution, which is the null hypothesis. Small p-values (e.g. $p < 0.05$) indicate the evidence is enough to reject it. In that case, the comparison of medians provides useful information. Table 7.1 shows the results from the test. Also, the table presents results of the χ^2 goodness-of-fit (GOF) test to investigate the normality of data. This result is presented because other methods could be better than the Kruskal-Wallis test when data are normally distributed (e.g. one-way ANOVA). In table 7.1, DJI and IPC denote the results obtained from the different data sets. The label “KW Test” indicate the results are the p-values obtained from the Kruskal-Wallis test for the corresponding experiment. The methods are denoted in the following way: Investment Strategies (IS), buy-and-holds (BH), single period portfolios (SP), investment strategies optimized using standard deviation (SI), and investment strategies optimized using value-at-risk (VI). The Kruskal-Wallis test can be applied to many samples simultaneously, but the results show the comparison of pairs of methods. For example, the label “IS/BH” indicates an investment strategies sample is compared against a buy-and-hold sample for the experiment of a particular performance metric. Table 7.1 shows the test results for every combination of metrics and pair of methods. Although, the stop loss occurrence metric was not included because further testing was not necessary in that case.

In general, the p-values from the χ^2 goodness-of-fit test indicate the data are not normal, which is somehow perceived in the box plots shown below. This justifies the use of the Kruskal-Wallis test. Besides, they indicate the maximum loss of single period portfolios from the IPC experiment (further explained below) could be normal. Nevertheless, the Kruskal-Wallis test was applied to all the cases.

Also, the results shown in table 7.1 indicate most of the distributions of the results are statistically different from each other. This means the difference of medians is also significant and their comparison can be used to draw conclusions form the experiments. The cases where the null hypothesis was not rejected are discussed in the following sections.

7.2 Discussion of Results

The results appear from figure 7.1 to figure 7.21. Box plots were chosen to display the results because they provide further information about the distribution of the data. Figure 7.1 shows

Table 7.1: Hypothesis Tests Results for Comparison of Methods

DJI KW Test	IS/BH	IS/SP	BH/SP	SI/VI
Maximum Loss	0.0000	0.0000	0.0000	0.9326
Total Periods	0.0008	0.0000	0.0000	0.7580
Final Return	0.0000	0.0000	0.0000	0.1509
Sharpe's Ratio	0.2631	0.6539	0.4093	0.3654
IPC KW Test	IS/BH	IS/SP	BH/SP	SI/VI
Maximum Loss	0.0177	0.0000	0.0000	0.9738
Total Periods	0.0000	0.0004	0.0000	0.0136
Final Return	0.4940	0.0000	0.0000	0.8862
Sharpe's Ratio	0.0000	0.0000	0.0140	0.0000
DJI χ^2-GOF	IS	BH	SP	
Maximum Loss	0.0000	0.0000	0.0000	
Total Periods	0.0000	0.0000	0.0000	
Final Return	0.0000	0.0000	0.0000	
Sharpe's Ratio	0.0000	0.0000	0.0000	
IPC χ^2-GOF	IS	BH	SP	
Maximum Loss	0.0000	0.0000	0.3248	
Total Periods	0.0000	0.0000	0.0000	
Final Return	0.0000	0.0000	0.0000	
Sharpe's Ratio	0.0002	0.0000	0.0000	
KW Test	DJIs/DJIv	IPCs/IPCv	DJI/IPC	
Duration	0.8851	0.0003	0.0000	
	DJIs	DJIv	IPCs	IPCv
χ^2-GOF	0.0007	0.0000	0.0000	0.0000

the maximum loss obtained by the methods for each random instance of the experiment. In the figures, SIs and VIs refer to investment strategies which used standard deviation and value-at-risk, respectively. IS denotes both types as a whole, BH refers to buy-and-holds, and SP is used to identify single-periods methods.

Figure 7.1 show ISs maximum loss was less than the losses obtained by B&Hs or SPs. The maximum loss suffered by the standard deviation ISs and the VaR ISs is similar. The three methods suffered heavy losses in some cases. This seems to indicate there were instances where heavy loss was unavoidable. Nevertheless, heavy loss was more common for B&Hs or SPs. This is indicated by the extension of their whiskers. Extreme loss (around 90%) was an outlier for every method. The difficulty of instances is investigated in figure 7.6. In this context, difficulty of instances refers to the probability the methods had to reach the goal in time. The box plots indicate a loss heavier than the stop loss is a common occurrence for about 40% of the instances. The lack of outliers in most of instances confirm heavy loss is more the rule than an exception. Cases 4, 16, and 30 proved to be the most difficult ones. They are probably the cause of the outliers appearing in figure 7.1. On the other hand, instances 5 and 20 presented small losses in a regular basis.

Figure 7.2 shows a similar behavior. The total number of periods to reach the goal is

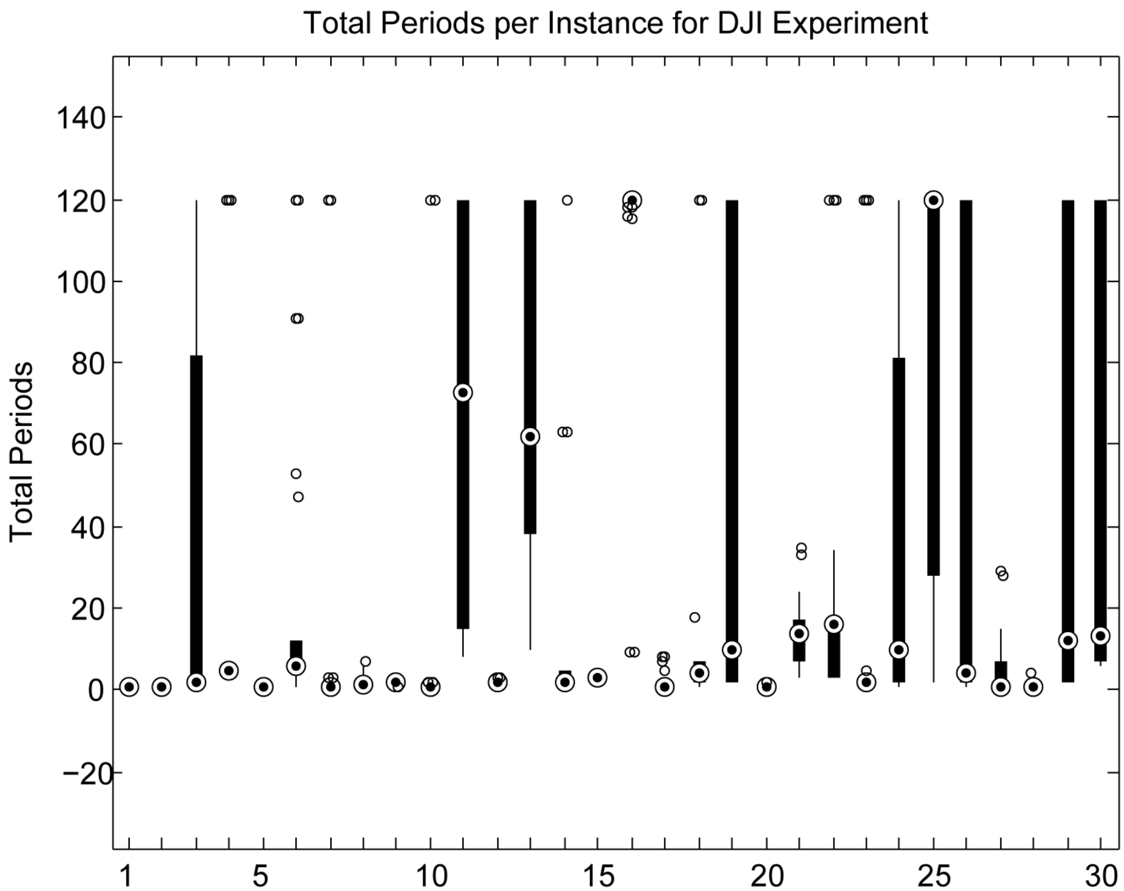

Figure 7.7: Total Periods for DJI Experiment per Instance

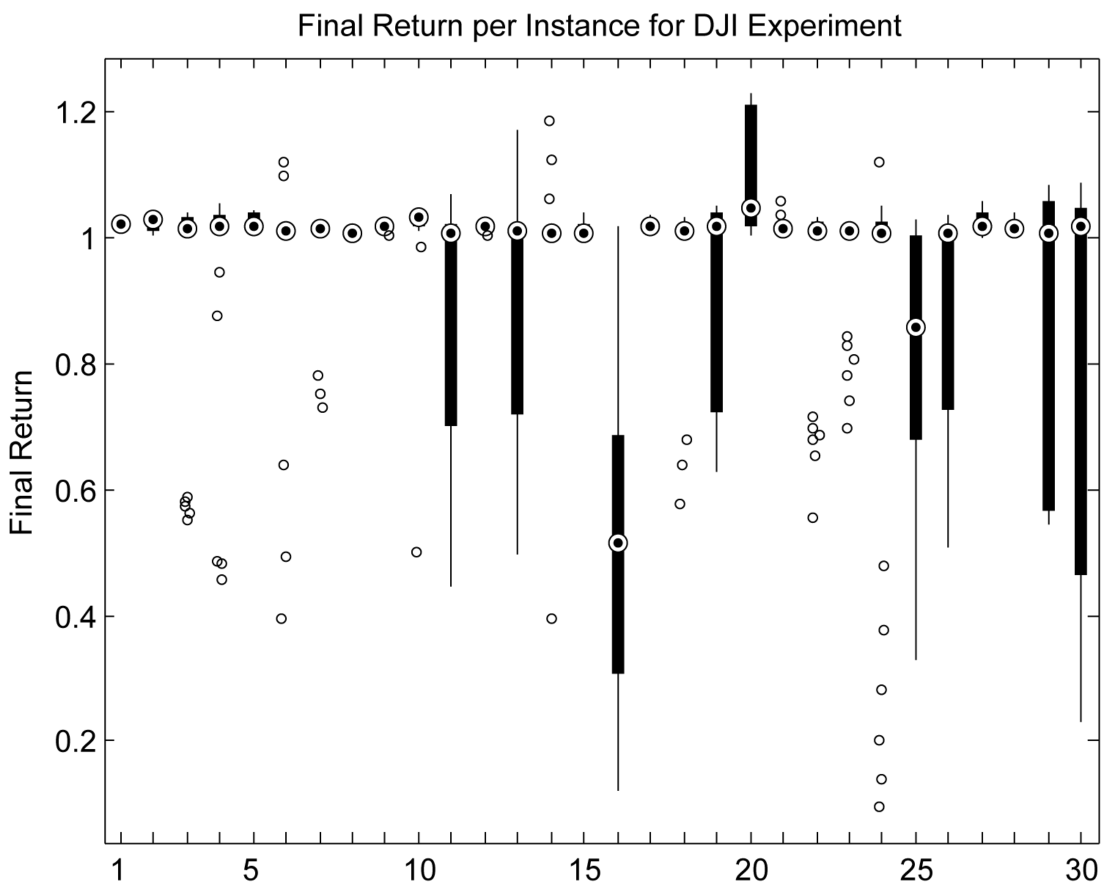

Figure 7.8: Final Return for DJI Experiment per Instance

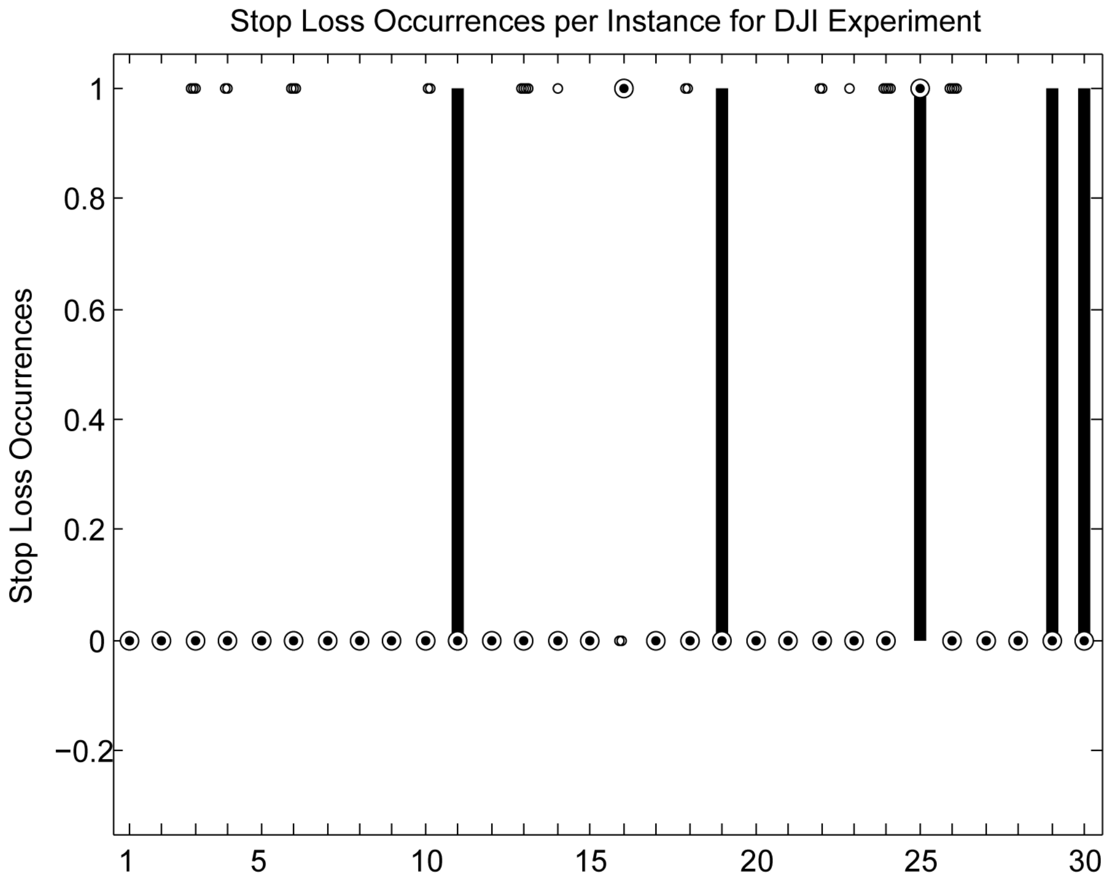

Figure 7.9: Stop Loss for DJI Experiment per Instance

shown in this one. The number of periods seems correlated to the difficulty of instances. Besides, a short set of portfolios seems to take less risk than a long one. ISs shows to have needed less periods than the other methods. Besides, ISs distribution is narrower than the other methods. Outliers are also present in this case for ISs and B&Hs. A extremely large number of periods was not a common case for ISs and B&Hs. This is contrary to the SPs case, where failure to reach the goal was more frequent. Figure 7.7 shows around 30% of the instances required large numbers of periods to solve them. The worst-case scenario was the norm for instance 16. Most of the methods failed to solved it and the fortunate cases were outliers.

Figure 7.3 shows the final return of the different methods. The termination of the run when the algorithm reached the goal avoided the occurrence of large returns. The goal was selected based on the investor's preferences captured by the interest rate, the initial number of periods, and the value of $\alpha_{\sigma M}$. More ambitious goals would require a revision of the investor preference. Figure 7.3 indicates ISs attained larger final returns than the other methods. Both SIs and VIs had a similar behavior. The distributions of ISs are located above 1, this indicates ISs usually attained profit. B&Hs have a narrower distribution, but they have a higher probability to attain negative return than ISs. SPs were prone to attain lower returns than the other

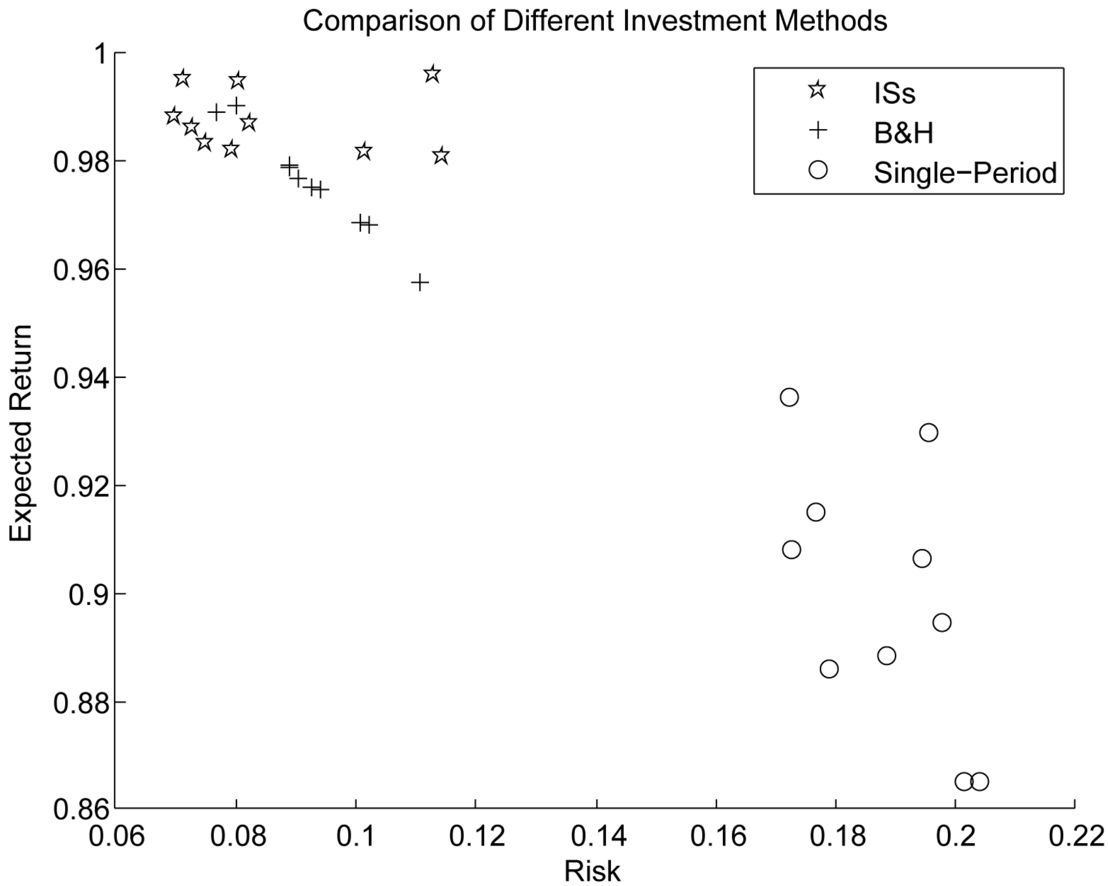

Figure 7.10: Pareto Front for DJI Experiment

methods. ISs and B&Hs presented outliers. There were instances where the methods attained an unexpected low return, but other where they attained an unexpected high return. This happened more frequently for ISs than B&Hs. Negative cases could be explained considering the difficulty of particular instances. There is the possibility the easy cases are the same for both methods, but ISs attained higher return than B&Hs for these particular instances. Figure 7.8 confirms instances 16 and 30 where the hardest ones. In general, this figure provides similar information than figure 7.6 and figure 7.7.

Figure 7.4 presents the number stop loss occurrences for each method. This figure shows stop loss was a common occurrence for SPs only. For the rest of the methods, this was an outlier. Figure 7.9 shows instances 11, 16, 20, 25, 29, and 30 are the ones where stop loss occurrences were common. The rest also presented some of them, but they were outliers.

Figure 7.5 shows the Sharpe's ratio obtained by each method. The median of the Sharpe's ratio is similar for SIs and VIs, but SIs has a broader distribution and this ones is biased to negative Sharpe's ratio values. This means the probability to attain a negative Sharpe's ratio using SIs, compared to VIs. SPs attained the highest median of the three methods, but their distribution is largely biased to negative values. The median of ISs was positive while the median of B&Hs was negative. This results seems related to the amount of information

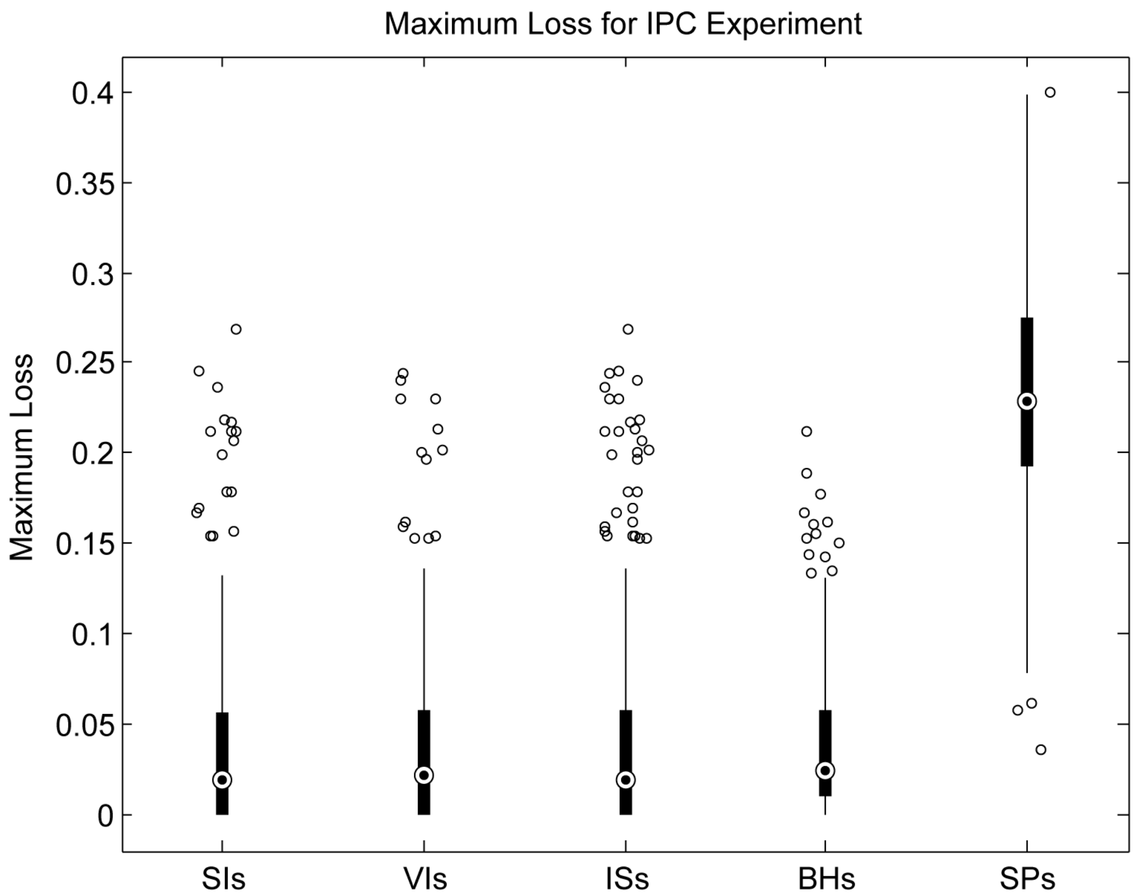

Figure 7.11: Average Maximum Loss for IPC Experiment per Method

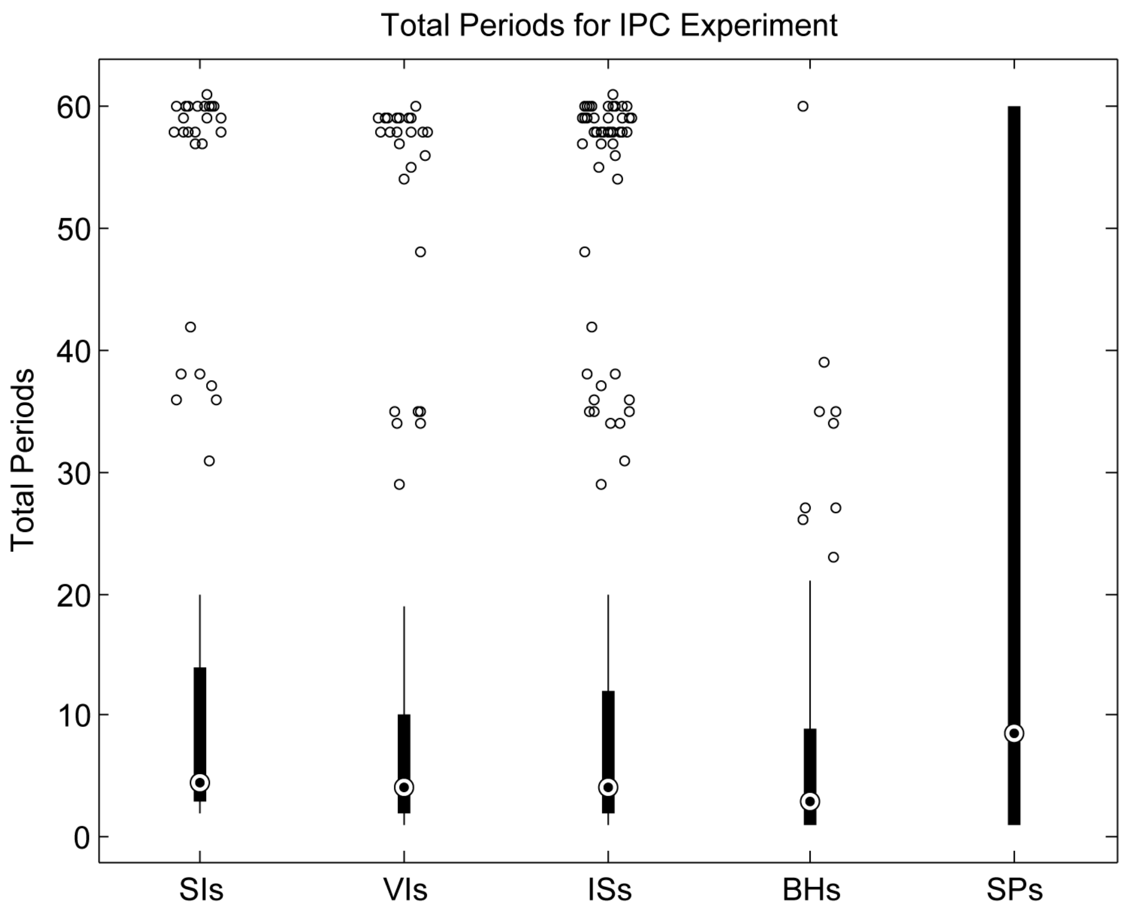

Figure 7.12: Total Periods for IPC Experiment per Method

used by the methods. For example, SPs use the largest amount of information because they do a portfolio revision each period. B&Hs are the contrary case, they never do a revision in favor of avoiding transaction costs. SPs seems to take less risk than other methods because of their revision frequency. Although, excessive revision seems to incur in unnecessary transaction costs in most of the cases.

According with table 7.1, there is no statistical evidence the Sharpe's ratio distributions of the methods are different. In that case, the difference of medias do not offer useful information. Nevertheless, McDonald (2015) indicated some references warns about the similarity of the sample's distributions and the validity of results (Fagerland & Sandvik, 2009). The compared distributions should have similar shape. For example, the fact distributions are skewed to different directions would lead to test errors. Also, the variances of the samples should be similar. Figure 7.5 shows the ISs distribution is approximately symmetrical while the others are skewed towards negative values. This difference could provoke the test fail to determine the medians of IS and the others are significantly different. On the other hand, it seems reasonable the medians of SIs and VIs are similar. The same could be said about the distributions of B&Hs and SPs.

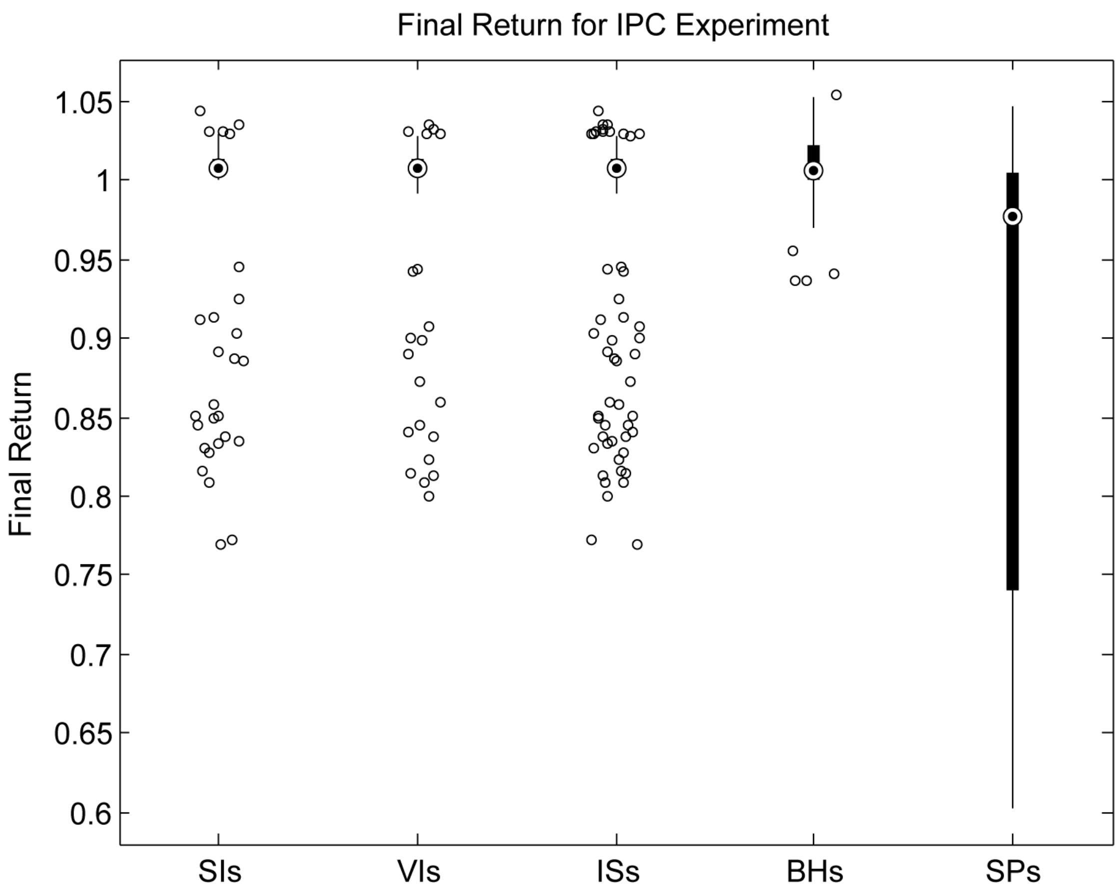

Figure 7.13: Final Return for IPC Experiment per Method

ISs are the middle point between SPs and B&Hs. They do portfolio revisions but at a lower frequency than SPs. On the other hand, ISs also consider transaction costs and include a fixed-return asset which can be used to go out the market in complicated scenarios. Nevertheless, this resource is only useful when good estimations are available. The abuse of the fixed-return asset will negatively impact the portfolio performance. This results seems indicate the balance between the use of information and the avoidance of transaction costs is crucial to develop effective investment methods.

In general, table 7.1 indicates there is no significant differences between the medians of SIs and VIs for this data set. This seems indicate the performance of investment strategies optimized with different risk metrics is similar.

Figure 7.10 presents the Pareto front of different methods. The average of risk and return of each method variation was plotted in the risk-return plane. The averages where computed along instances for each possible variation. For example, the risk and return of the investment strategy S_1 was computed for each instance and their averages were plotted in figure 7.10. The figure indicates the Pareto front of policies is exclusively composed by ISs. Both ISs and B&Hs dominate SPs in this case. The average and medians of distributions have not necessary the same value. Discrepancies occur when large differences in data are present.

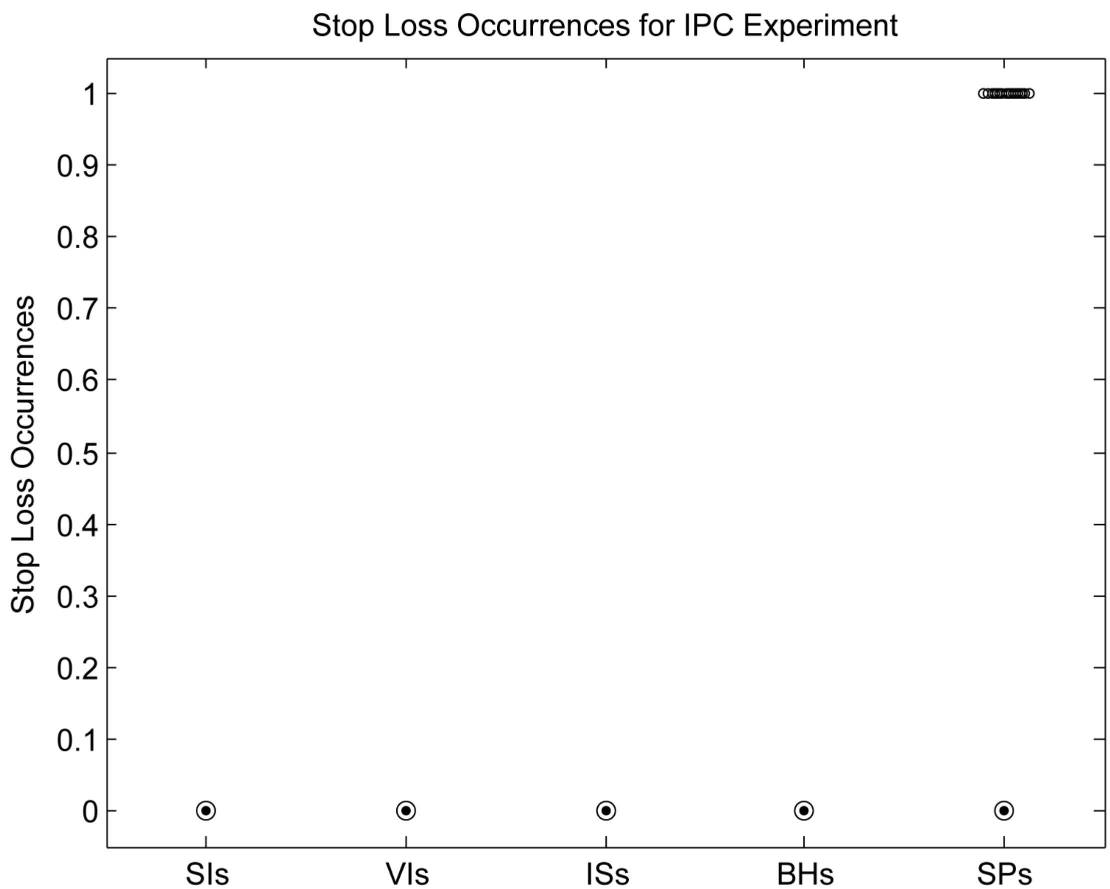

Figure 7.14: Stop Loss for IPC Experiment per Method

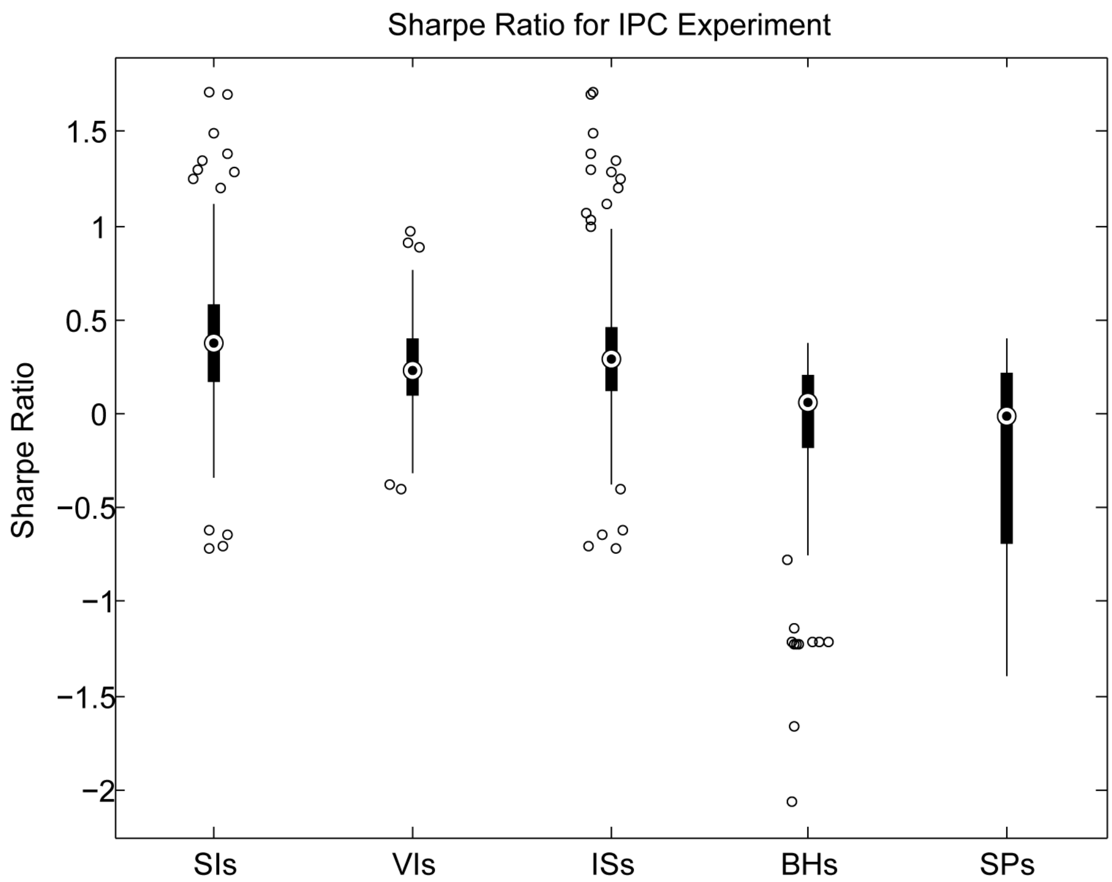

Figure 7.15: Sharpe's Ratio for IPC Experiment per Method

A similar analysis could be done for the IPC experiment. Figure 7.11 shows the maximum loss per method. SIs and VIs showed a similar performance. The median of ISs is lower than the other methods. Table 7.1 indicates the difference is significant for a critical value of 0.05. Small losses are more probable with ISs than B&Hs. There is not a low whisker for B&Hs box plot. SPs maximum loss was significantly higher than the one of other methods. In this cases, losses in the range of the other methods are outliers. Losses higher than 15% were outliers for both ISs and B&Hs. In general, the behavior is similar to the one shown in figure 7.1 for the DJI experiment. Nevertheless, losses were much lower for the IPC experiment, where 40% was the lowest reported value. Figure 7.16 provides further information among both experiments. The DJI experiment shows more extreme cases (both hard cases and easy cases) than IPC experiment. Figure 7.16 indicates all the cases have a similar probability to attain a particular loss level. Cases 15 and 30 are the ones with higher median.

Figure 7.12 shows the total number of periods per method. SIs and VIs have similar performance, but SIs have a broader distribution bias to higher number of periods. B&Hs is the method with the lowest median and its followed by ISs. Policies larger than 20 periods were outliers for ISs and B&Hs, while the maximum possible is a common occurrence for SPs. Based on figure 7.2, the distribution of ISs and B&Hs was similar for both experiments.

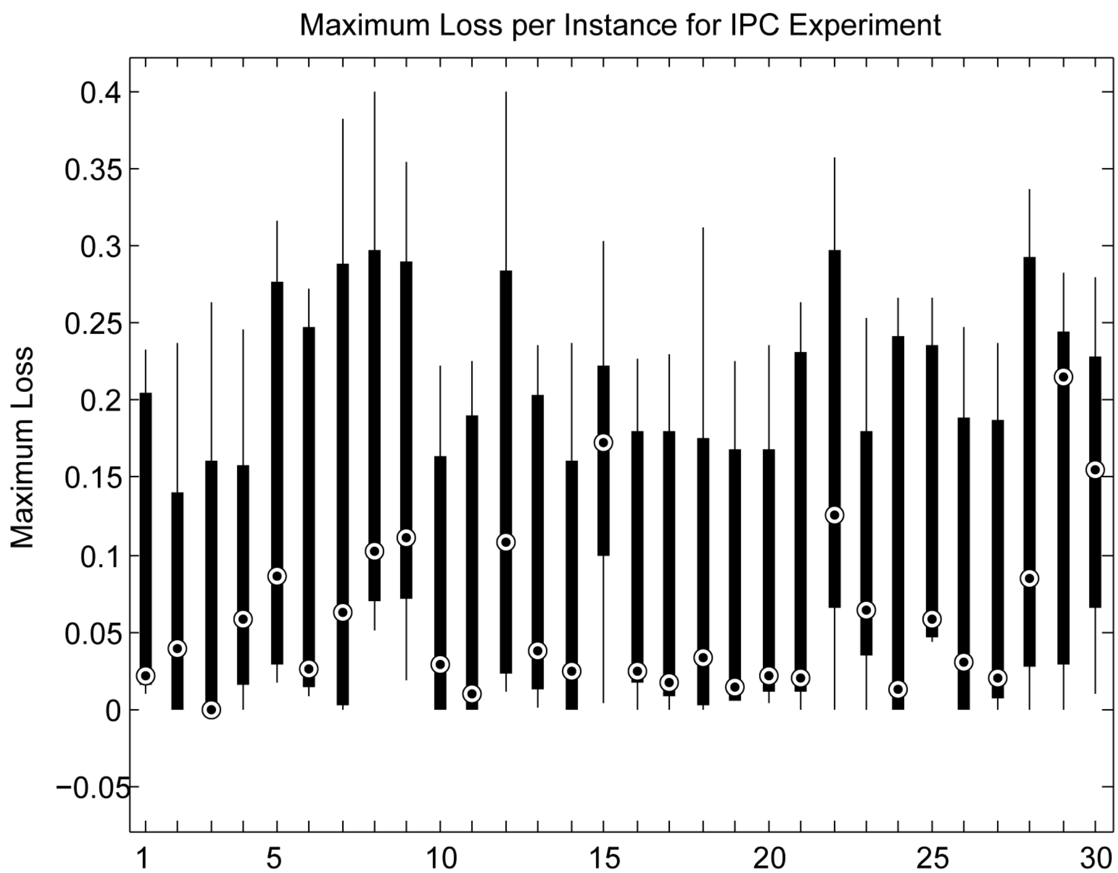

Figure 7.16: Average Maximum Loss for IPC Experiment per Instance

Figure 7.17 indicated about 50% of the cases were easy for the methods. Having a number of periods larger than 10 periods is an outlier for those cases. The contrary occurred for the rest of the instances.

Figure 7.14 shows the stop loss occurrences for the IPC experiment. In this case, stop loss seldom occurred. Only SPs reports occurrences, and they were outliers. Figure 7.19 confirm these results. Stop losses were outliers when they had place.

Figure 7.13 shows the final return per method. SIs and VIs have similar median, but SIs distribution is biased towards positive returns, compared to VIs. The median of ISs is higher than B&Hs and SPs. The distribution of ISs is narrower than the distribution of B&Hs. This indicate the probability of having negative returns is higher for B&Hs than for ISs. Table 7.1 indicates the difference is significant for a critical value of 0.05. SPs are clearly biased towards negative returns, where the lowest value was 60%. The behavior of the different methods was similar in both experiments. Nevertheless, the DJI experiment showed returns spanning the range of 10% to 120%, while the IPC experiments were in the range of 60% to 105%. Although, most of the extreme cases are outliers. This seems indicate more extreme and unpredictable scenarios occurred in the DJI market in those years. Figure 7.18 shows negative returns are likely for about 40% of the instances.

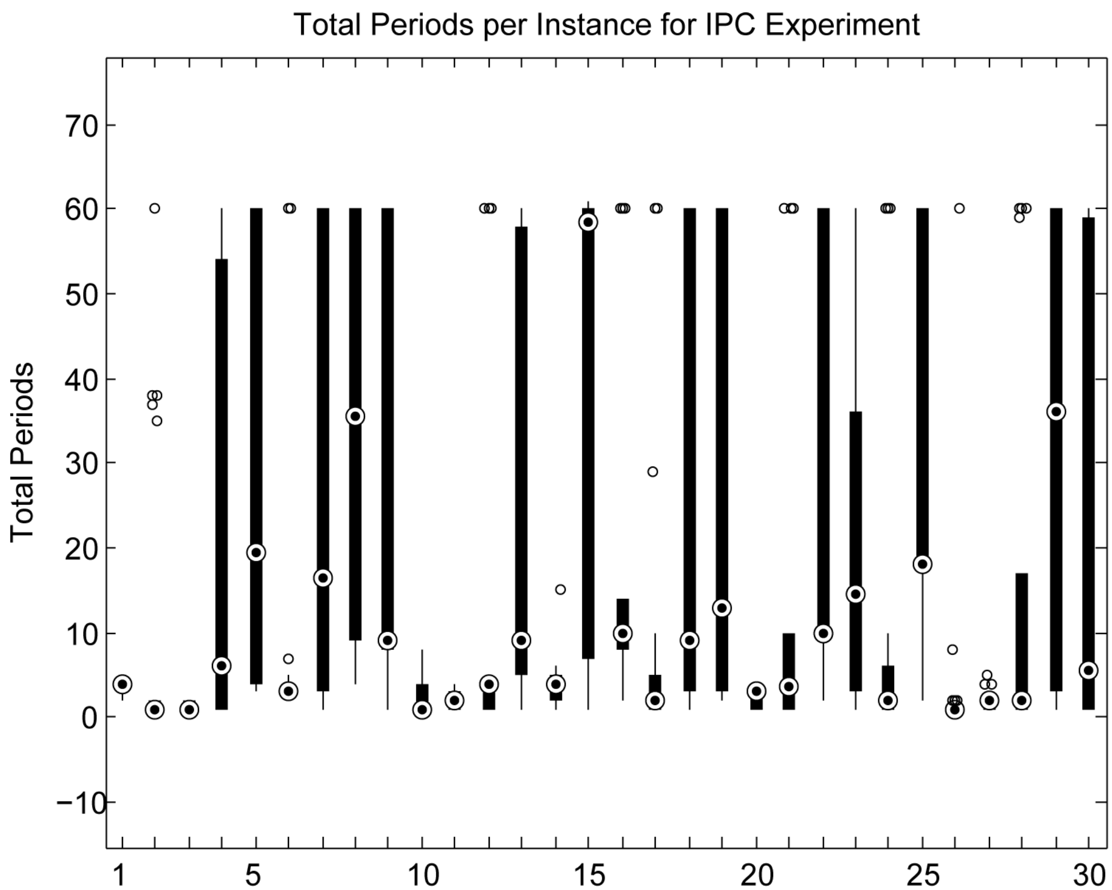

Figure 7.17: Total Periods for IPC Experiment per Instance

Figure 7.15 shows the Sharpe's ratio for the different methods. SIs have larger Sharpe's ratio median than VIs and their distribution are biased towards higher values. ISs attained a higher median than B&Hs and SPs. ISs distribution is more symmetrical then the others. Distribution of B&Hs and SPs are biased towards negative values. Although, the probability of negative values is higher for SPs than for B&Hs. Table 7.1 indicates the difference of medians between B&Hs and SPs is significant for a critical value of 0.05. The medians of the IPC experiment are higher than the ones of the DJI experiment. Although, the behavior of the methods is similar in both experiments. These results seems indicate less risks was taken in the IPC experiment than the DJI experiment. This is concluded because figures 7.3 and figure 7.13 show similar final returns but the Sharpe's ratio of the IPC experiment were generally higher than the ones of the DJI experiment. Lower risk levels should have increased the Sharpe's ratio values in the former case. The fact SPs attained the lowest median is contrary than the DJI experiment, where they attained the highest median value. Nevertheless, the distribution is biased towards negative values in both cases. On the contrary, the performance of ISs and B&Hs increased in the IPC experiment. Given IPC market seemed to be less risky than DJI market, this seems indicate both ISs and B&Hs performed better in a low risk scenario.

In general, table 7.1 indicates there is no significant difference between SIs and VIs for any metric but the Sharpe's ratio. In that case SIs promised higher returns with lower risks than VIs. Standard deviation could be regarded as optimistic compared to VaR because it considers total risk instead of downside risk. In other words, investment strategies optimized using standard deviation consider the final outcome could be better or worse than expected, while the ones optimized using VaR consider negative possible outcomes only.

Figure 7.20 shows the Pareto front of the average risk-return for the different methods. In this case, the Pareto front is composed by ISs only. Besides, there is a clearer difference among the methods than for the DJI experiment: The variations of methods seemed clearly clustered together. Both ISs and B&Hs dominate SPs.

Finally, figure 7.21 compares the number of periods a portfolio was hold by the algorithm before revision. DJIs and DJIv denote the investment strategies applied to the DJI experiment optimized with standard deviation and value-at-risk, respectively. The same applies for the IPC experiment. This figure indicates the algorithm applied sets of portfolios of different number of periods each time, which is a proof of the co-existence of them at the same multi-period Pareto front. The medians of the portfolios sets optimized using VaR is higher than the ones optimized using standard deviation, although table 7.1 indicates the difference is not significant for the DJI experiment. On the other hand, the differences for the IPC experiment are significant. In that case, the median of VaR ISs is higher than standard deviation ISs. Also, the distributions of VaR strategies are broader. This indicates the algorithm had a tendency to choose larger portfolios sets when VaR was considered as the risk measure. Moreover, The medians of the DJI experiment are higher than the medians of the IPC experiment, which is supported by the results of table 7.1. This means the algorithm chose shorter portfolio sets in the IPC experiment. The DJI experiment showed some outliers, where the largest set had 24 portfolios. This could be explained from the fact the methods suffered heavier losses in the DJI experiment than in the IPC experiment. According with a Martingale strategy, a gambler would need an infinite amount of money to win every bet (Finkelstein & Whitley, 1981). In a similar manner, a simple strategy to deal with loss is waiting for the market to recover by itself. Therefore, the number of period of a set of portfolios seems to be related to the losses

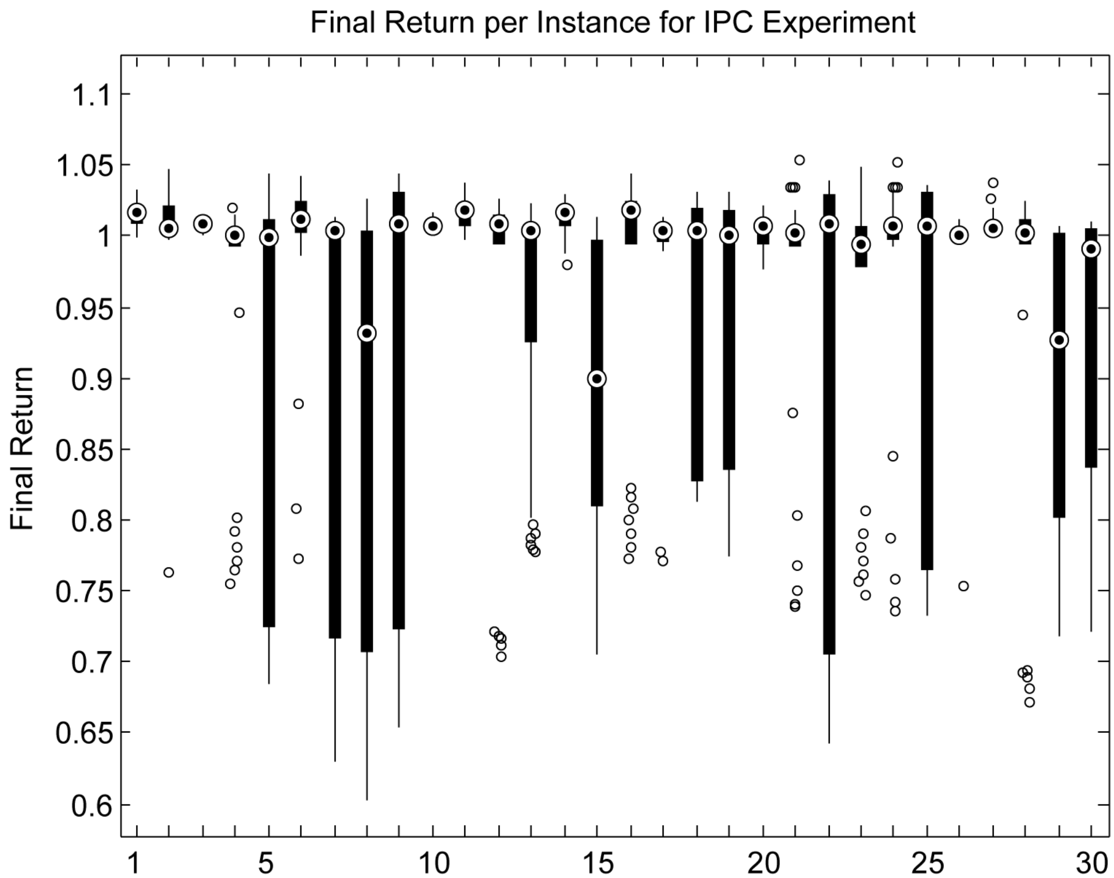

Figure 7.18: Final Return for IPC Experiment per Instance

suffered during the run. One possible reason for these loss scenarios is the economic crisis the United States suffered at 2008. The data covers this period of time.

The difference between markets is further investigated using hypothesis testing. The Kruskal-Wallis test was used to compare results of the experiments from each data set. In this case, the same pair of method and metric is compared from both markets. The p-values are shown in table 7.2. The results show the performance is different for all the combinations but the Sharpe's ratio of the single period portfolios. This supports the conclusion the American market had higher level of risk than the Mexican market.

Table 7.2: Hypothesis Tests Results for Comparison of Markets

Metric	IS	BH	SP
Maximum Loss	0.00202	0.00000	0.00000
Total Periods	0.00000	0.00000	0.00000
Final Return	0.00000	0.00039	0.00317
Sharpe's Ratio	0.00000	0.00003	0.10897

The presented results seems indicate ISs have better performance for the presented measures. In the DJI experiment, they showed to have lower maximum loss than other methods. Besides, ISs took less time to reach the goal and attained higher final return. Stop loss seldom occurred. The probability distribution of ISs Sharpe ratio has higher probability to attain positive values than the ones of other methods. The results are similar for the IPC experiment, although the median of total periods was higher for ISs than for B&Hs. Finally, the Pareto front of the average risk-return of the methods was composed by ISs only. In general, all three methods showed similar behavior in both experiments. The less effective method was SPs, which seems a prove neglecting transaction costs and other dynamic restrictions cannot produce optimal investment decisions.

The results also indicate the DJI experiment presented a higher level of uncertainty than the IPC experiment. For example, maximum loss and stop loss occurrences were higher in the DJI experiment. This was explained by the economic crisis American markets suffered in 2008, which is a period of time included in data. Also, there were found instances where heavy loss was unavoidable regardless the method.

The effect of considering different risk measures was investigated. The result indicate VaR ISs have a tendency to be generate larger sets of portfolios. This implies risk increase more slowly with the number of periods when VaR was considered. In other words, the increase in risk from sets of portfolios with T periods to sets with $T + 1$ periods was more drastic when the standard deviation was considered instead of VaR. This could be from the fact the probability distribution of final return is not symmetric. VaR considers downside risk while standard deviation considers total risk. Finance theory considers investor is not concerned by upward risk, which is regarded to be beneficial to investment. Although, there was not found a significant difference between the performance of both standard deviations ISs and VaR ISs. Although, the optimal portfolios found using each risk measure were different from each other. The advantages of using different risk measures should be further investigated.

Finally, it was found the method truly used policies with different number of periods at each revision time. This is a proof a multi-period Pareto front could be composed by sets

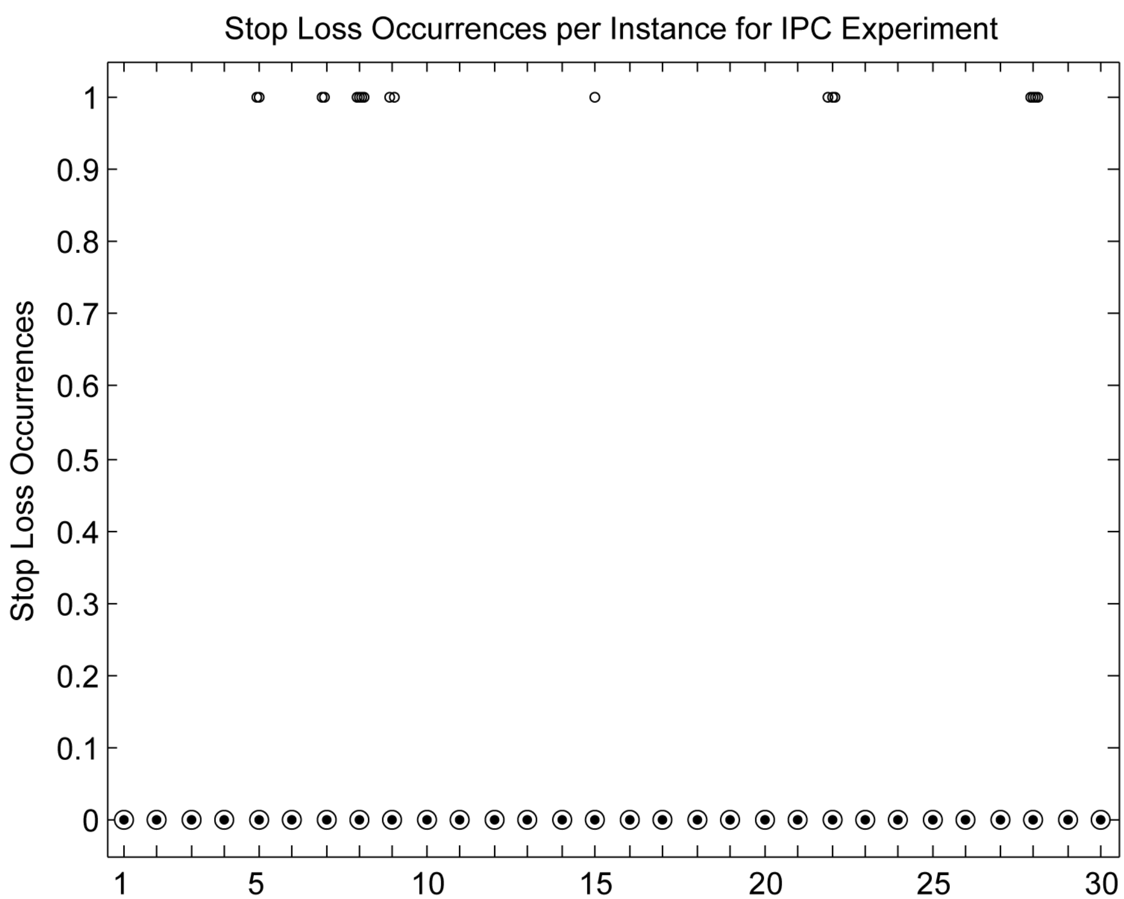

Figure 7.19: Stop Loss for IPC Experiment per Instance

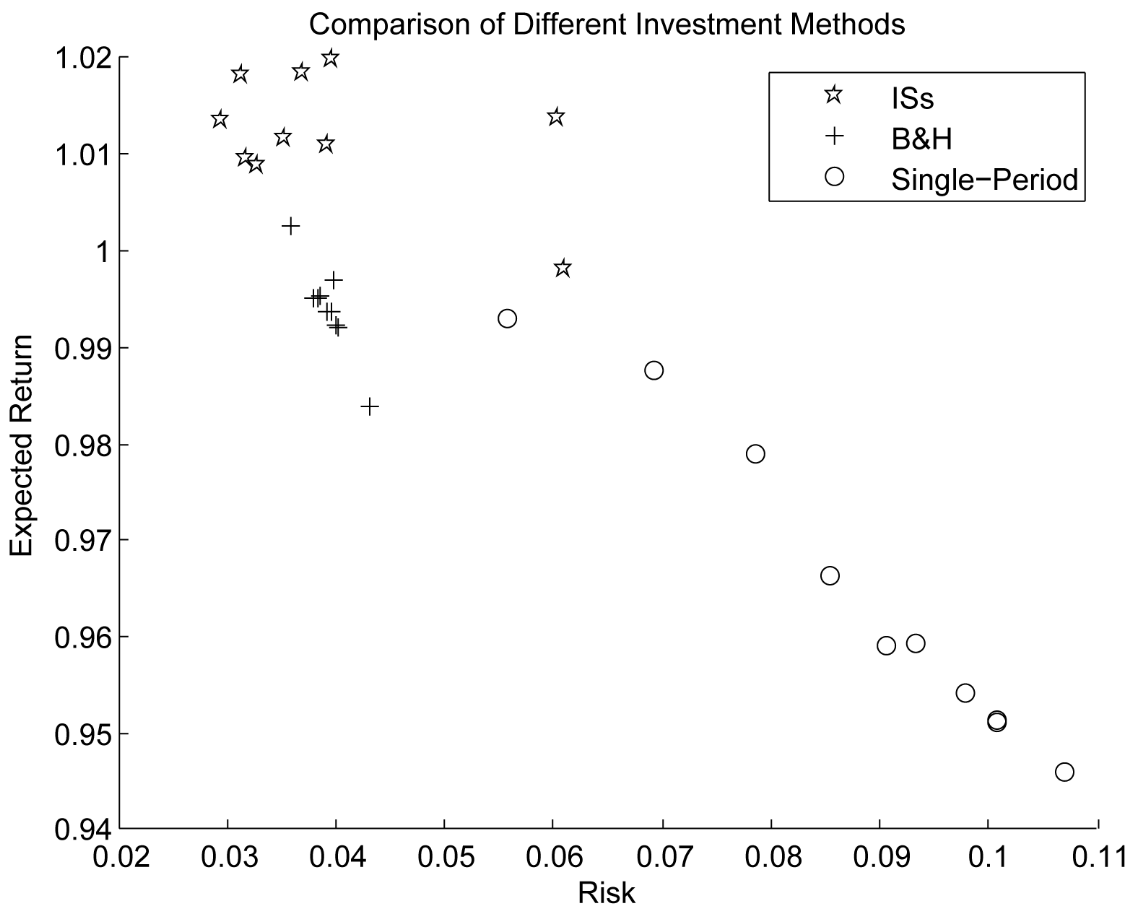

Figure 7.20: Pareto Front for IPC Experiment

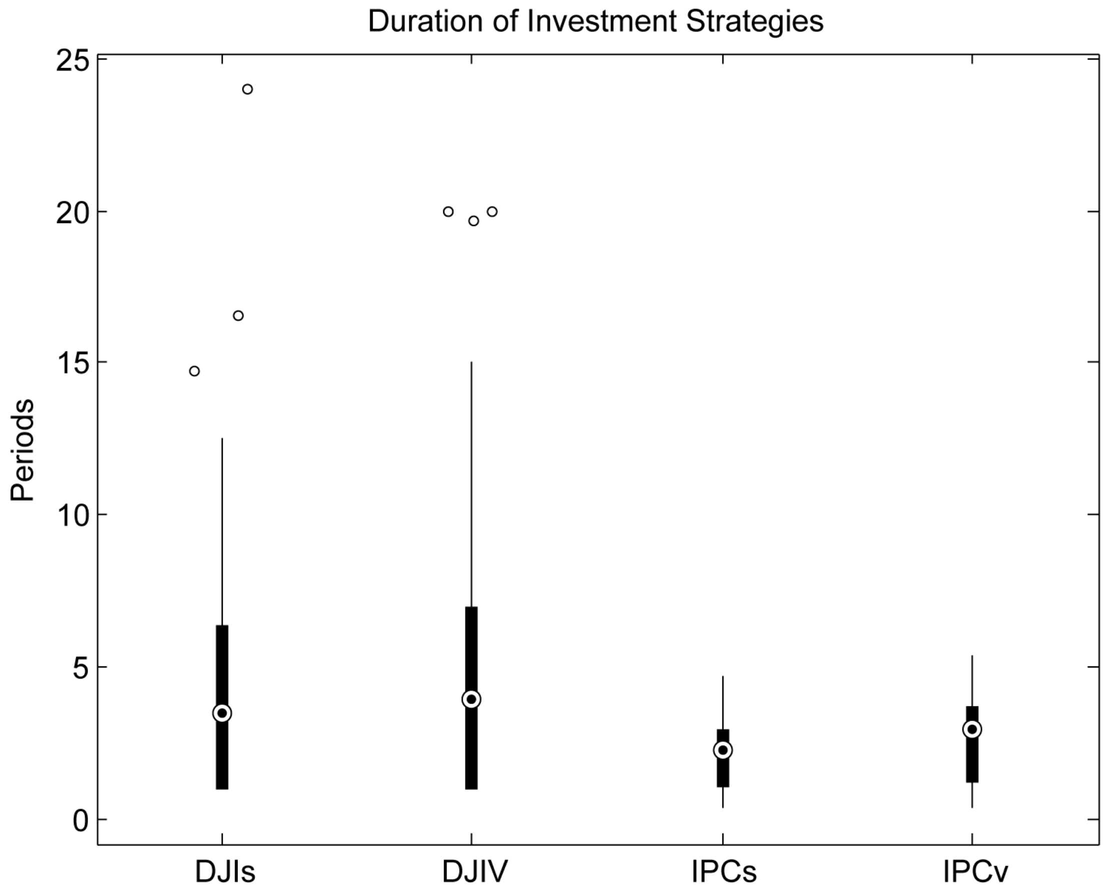

Figure 7.21: Duration of Different Investment Strategies

of different periods. These Pareto fronts provide investors with more flexible options at the moment of decision making.

7.3 Conclusion

This chapter presented the results of experiments conducted to investigate the performance of the proposed method and the nature of markets. The chapter described the experiments and the performance measures in the first part. The proposed measures were the following: Maximum loss, total periods, final return, stop loss occurrence, Sharpe ratio, and domination. The following section showed the results for the DJI experiment and for the IPC experiment, respectively. Box plots were used to display the information about the probability distribution of the results. Besides, hypothesis testing was used to support the conclusions obtained from the box plots analysis. They showed ISs had better performance than other methods for the considered performance measures. A higher level of risk was identified in the American market. The economic crisis was considered one of its possible causes.

It was concluded that different risk measures could lead to different optimal portfolio policies. Although, significant difference in performance among them was found. The main difference was VaR polices have a tendency to find longer policies than standard deviation investment strategies. The asymmetry of the distribution of final return was one of the possible explanations to this phenomenon. Finally, the existence of multi-period Pareto fronts composed by portfolio sets of different number of periods was proved in the results.

Chapter 8

Conclusions

This is the final chapter. The first part is a summary, the second part intends giving answer to the research questions proposed on the first chapter. The next presents some final comments and general conclusions. The last one is the future work section.

8.1 Summary

This work proposed a new evolutionary computing method based on portfolio theory to make investment decisions. Portfolio theory was preferred because it includes the risk of the decision into the optimization process. It acknowledges our inability to make a perfect prediction of the future and provides solutions considering the outcome could be worse than expected. From that perspective, portfolio optimization is similar to robust optimization. Although, there are differences which make risk optimization interesting on its own terms.

The research took two complementary directions: The first part studied the limitations of the traditional portfolio theory and the way to overcome them. The second studied the application of evolutionary algorithms to solve financial problems. The first study showed traditional portfolio theory excluded dynamic restrictions from the problem definition and relied on theoretical utility functions to explain how the investor chooses one portfolio from all the available possibilities. These limitations diminish the effectiveness of portfolios in real-world investment. Both investors and finance professionals have looked for methods to solve these problems. The literature reported multi-period definitions of the problem to allow the introduction of transaction costs and other dynamic restrictions. Nevertheless, the solution of multi-period portfolio problems proved to be difficult from a mathematical framework. Also, the difficulty of the problem limited the exploration of other possibilities, like the inclusion of data innovations or identification of the investor's preference.

The second study showed that multi-objective evolutionary algorithms have been mainly used to solve portfolio selection problems with static restrictions. They are a natural choice for this problem because of their ability of simultaneous optimization of risk and return. On the other hand, the solution of portfolio selection problems with dynamic restrictions had received limited attention in the literature. This occurred because the references about multi-period portfolio selection are mainly concerned with finding closed-form solutions to the problem, while evolutionary algorithms are better suited to find numerical solutions. Nevertheless, this latter approach has the advantage to include dynamic restrictions in a simple manner.

The conclusion was an evolutionary computing method is a good approach to solve multi-period portfolio problems with dynamic restrictions. An investment method based on multi-period portfolio theory could now be developed given the existence of an algorithm to solve multi-period portfolio problems. The method could explore all kind of restrictions, but the scope of this work was limited to transactions costs, unbalance, inflation, and the inclusion of data innovations. It also includes identification of the investor's preference.

The solution model can be divided in two parts: The first part is the multi-objective evolutionary algorithm to solve multi-period portfolio problems. The other is the investment method developed from the multi-period portfolio theory. The investment strategies method was developed to make investment decisions which considered dynamic restrictions. The hypothesis stated their inclusion could help making better decisions.

The experiments showed this approach had better performance than buy-and-holds and single-period portfolios, which are common practices of investors. Risk-weighted measures were considered to compare the different methods. Besides, the experiments concluded the existence of multi-period efficient frontiers with heterogeneous time horizons. Also, the results identified differences about the nature of Mexican and American markets. The use of different measures of risk leaded to different optimal sets of portfolios.

8.2 Answers to Research Questions

On the first chapter, the problem statement was further developed into a list of research questions. This section intends give an answer to them based on the results obtained from the experiments. The questions list is shown below.

- How can the investor's preference be included in the optimization process?
- Can utility functions be excluded from the portfolio selection problem?
- Which parameters could capture the information represented by utility functions?
- Is there inherent differences between the nature of different markets? How do the affect the optimization process?
- What differences among markets are instrumental when performing multi-period portfolio optimization?
- How can an evolutionary algorithm be implemented to solve multi-period portfolio selection problems?
- How should be measured the performance of financial strategies?
- How the proposed method could be appealing to regular investors?

This work showed the investment strategies method includes the investor's preferences by defining a goal to the optimization. An iterative procedure was proposed to help the investor to define his own preference based on the evidence provided by the multi-period Pareto front, which is computed from the current state of the market using a Monte-Carlo approach. The goal of return and risk is defined based on the information provided by the front. The next multi-period Pareto front is computed based on the previous decision, which is made based on the goal. In the experiments, the algorithm stops when the portfolio value fulfill the investor's expectations, but the process could continue as long as arriving data are available. This work limited the research to fixed goals, but it is possible to define new goals based on the market state and the portfolio's performance. This could be reasonable because the market is assumed to be non-stationary, therefore, the investor could aspire to better outcomes under different market conditions. On the other hand, bear markets could turn the original investor's expectations into unreasonable demands. Although, definition of adaptive goals remains an open question.

The second point inquires about the possibility of identifying the investor's preference instead of using theoretical utility functions to describe it. Utility functions make the assumption the dynamics about preference is the same for all the investors. On the other hand, other fields (e.g. automatic control engineering) use identification process regularly. In those cases, general models have numerical parameters which are adjusted based on data from the particular system. This information is crucial to design effective control rules for them. In this work, this idea was extended to the investment framework. Nevertheless, the evolutionary algorithms approach is able to implement utility functions to define the investor's preference. A comparison of both approaches could draw interesting conclusions.

The third question is about the way the identification process should be done. The investment strategies method defines the investor's preference based on the maximum number of periods from the initial Pareto front ($T_{\max}$), the maximum risk assumed by the Pareto front ($\sigma_{\min}$), the fraction of maximum risk the investor's is willing to take (α) and the minimum return he is willing to achieve ($X_{T\min}$). The goal is defined by the risk and return of the portfolio with closer risk level to $\alpha\sigma_{\max}$. The goal and the maximum number of periods are the most significant parameters. The goal is considered in all the investment decisions made by the algorithm. On the other hand, $T_{\max}$ captures the relationship between the investor's preference and the state of the market. More demanding goals will take longer times to be accomplished. Therefore, $T_{\max}$ will affect the average number of periods the algorithm will hold the portfolios before revision.

The following point is about the nature of different markets. The experiments compared indexes from the American market and the Mexican market, respectively. The results indicated the level of risk of the American market was higher than the risk of the Mexican market. The American market showed more dramatic losses than the mexican market. Also, the algorithm preferred longer revision times at the American market experiments. These results seem indicate waiting is a valid investment decision when the risk is high. The results also indicated waiting is a better strategy than continuous (probably misguided) changes to the portfolio. The reason is unnecessary portfolio's changes incur into excessive transaction costs. Finally, these results indicate the possible investment decisions are the same in both markets, but information about their current state is useful to choose the best option for a given situation.

How can an evolutionary algorithm be implemented to solve multi-period portfolio selection problems? The solution model considers the investment strategies method to make investment decisions based on multi-period Pareto fronts including information about the market state, current performance and the investor's particular preference. Also, an evolutionary algorithm to perform multi-objective risk optimization is presented in this work. The algorithm is used to compute the multi-period Pareto fronts needed to implement the investment strategies method. The algorithm includes a Monte-Carlo approach which allows to introduce inflation, transaction costs, portfolio unbalance, and other restrictions to the optimization process.

How should be measured the performance of financial strategies? This question is explored on chapter 6. Traditional finance theory measures risk and return of the portfolio based on the time series of their actual returns along time. This method estimates the distribution of the final return of the portfolio based on the distribution of the time-series. Nevertheless, the method is not specially suited to compute the final portfolio's risk (i.e. volatility). Finance theory acknowledges this fact and provides more sophisticated approaches to estimate the portfolio's volatility. For example, GARCH models can be used for this end. In this work, the Monte-Carlo approach was used instead to estimate the distribution of final return of the studied methods, allowing the estimation without the encountering the complications of other approaches. In general, this is the advantage of evolutionary algorithms. Moreover, that chapter discusses some of the limitations of using market indexes as benchmarks to measure the performance of investment methods.

How the proposed method could be appealing to regular investors? Some authors have acknowledged some investment methods are beyond the knowledge of regular investors, therefore, their advantages are overlooked by the public. The solution model proposed in this work has the advantage to be fully automatic, exempting the investor from the hard decisions he should make when managing his portfolio. The solution model provides a method to select the most suitable securities and provides decisions based on the investor's preferences, market state, and current portfolio performance. Also, the results seems indicate the proposed method allows making decisions with higher returns and lower risk. Nevertheless, further testing is needed to provide full evidence of this claim. More application of the method to real investment situations is needed to provide the investors with figures about the performance. On the other hand, the proposed method was found to make conservative decisions, because the inclusion of the market state prevents the algorithm to set unreasonable goals. This fact should provide some confidence to investors because the method does not make promises which cannot be fulfilled for the current market conditions. On the other hand, the method's decisions could be presented to the investor as recommendations, where the investor could have the possibility overdrive the algorithm's decisions. Some algorithms [e.g. EDDIE (E. P. Tsang et al., 2000)] work under that approach.

8.3 Final Comments

The method devised in this work used the information provided by the data history in a different manner than simple-period approaches. The multi-period model allowed the method compute portfolio sets with different number of periods, while the single-period method makes a portfolio revision at each period. This means single-period methods use all the information available from the price data. Nevertheless, it attained the lowest performance in the experiments. On the other hand, the buy-and-hold method allow the portfolio to drift with the trends of the market. It uses the minimum possible information from the data, but it explodes the trends to attain profit and avoid transactions costs.

The proposed method exploited information from three different sources. The first one is the market itself. The method considered transaction costs, unbalance and inflation. All of them can be seen as properties of the market. Therefore, the method considered information from the market's nature into the optimization. The second source is price data, which is the same one used by the single-period method. Nevertheless, the investment strategies method use it to make estimations further into the future and for evaluation of the current state of the portfolio (i.e. data innovations). Finally, it takes advantage of the market trends like the buy-and-holds do. The time horizon of the portfolio is an estimation about the market trend. Therefore, information from trends is also exploited by the investment strategies method.

The conclusion from this analysis is the following: Information is crucial to make good investment decisions, and it is better to use information from the maximum number of sources possible. The poor performance of the single-period method could be attributed to it considered information from one source only. In other words, it used all the information from the price data but ignored the current state of the market and its trends.

The first chapter discussed the ideas of extraordinary profit and market efficiency. The results showed cases of both extraordinary loss and some scare cases extraordinary profit (compared to the average value). Although, these fortunate cases seemed to be random and non-predictable. In that case, it would not be possible to devise a strategy to exploit them. That result favors the efficient market hypothesis. Although, the experiments stopped when the portfolio reached the goal, therefore, there is no way to determine if the results truly showed extraordinary profit. This is still an open question.

The technical approach is based on the idea the data history holds all the information about a security price, but the experiments indicated some scenarios were not predictable. There are two possible explanations: These scenarios could be predicted from other information sources besides data history. This would contradicts the technical approach. The second one is these scenarios are predictable with a further analysis of the data. The present method did not consider volume information or any technical indicators. This matter needs to be further investigated.

The decisions made by the method could be described to be conservative. The goal was set based on the market conditions and the investor's preference. In the original algorithm, evidence is provided to the investor to help him determine a reasonable goal. If the investor decides the return's goal is not enough for him, this means he should be willing to take higher levels of risk to attain it. In other words, the method only provides goals which are attainable given the current market conditions. The method could be applied several times to accumulate any desired level of wealth. This approach seems indicate profit can only be attained with time and effort. Nevertheless, the case where the portfolio's value is higher than the goal was not fully investigated in the experiments. Further investigation is needed to have conclusions about this favorable scenario and its relationship with final return. New investment approaches could be devised from that analysis.

Finally, the investment strategies method allowed emulation of different behaviors. For example, the method behaves like a Martingale when the investment strategy chooses portfolios based on its return only: It will choose the highest return portfolios when the current wealth is less than the goal, but it will choose the lowest risk portfolios in the contrary case. The opposite occurs when risk is the only factor considered by the investment strategy. The behaviors observed by the method are not that different from behaviors observed in investors, but the difference in performance would come from the information available to the method and its to use of it. The conclusion is a good investment method should be wise enough to decide when to take action and when to let itself to "sail freely with the wind of change".

8.4 Future Work

The method is still open to improvement. It could include other sources of information (volume, news, etc). The experiments showed more information is better when making decisions, but it should be used carefully. Also, static restrictions could be included as well as trading execution capabilities.

Another improvement is the use of other techniques to model the future behavior of returns. A multi-variate distribution was used in this work to simulate different scenarios and estimate the distribution of the individuals. Finance theory has studied GACRH processes to model the volatility of time-series. A combination of GARCH with evolutionary algorithms could be the next stage of the proposed system.

Finally, this work considered the investment strategy was fixed along the run. It was shown how the behavior of the system changes with the selected strategy. Therefore, there could be cases where a particular strategy is better than the others. A method to update strategies dynamically is open to further research.

References

Adebiyi, A., & Ayo, C. (2015). Portfolio selection problem using generalized differential evolution 3. *Applied Mathematical Sciences*, *9*(42), 2069–2082.

Aguilar-Rivera, R., Valenzuela-Rendón, M., & Rodríguez-Ortiz, J. (2015). Genetic algorithms and Darwinian approaches in financial applications: A survey. *Expert Systems with Applications*, *42*(21), 7684–7697.

Aizawa, A., & Wah, B. (1994). Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, *2*(2), 97–122.

Almgren, R., & Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, *3*, 5–40.

Andriosopoulos, K., & Nomikos, N. (2014). Performance replication of the spot energy index with optimal equity portfolio selection: Evidence from the UK, US and Brazilian markets. *European Journal of Operational Research*, *234*(2), 571–582.

Araújo, R. d. A., & Ferreira, T. A. (2013). A morphological-rank-linear evolutionary method for stock market prediction. *Information Sciences*, *237*, 3–17.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–part II: Soft computing methods. *Expert Systems with Applications*, *36*(3), 5932–5941.

At&t historical quote. (2015, February). Retrieved from `http://att.centralcast.net/historic_att_stock/`

Back, B., Laitinen, T., & Sere, K. (1996). Neural networks and genetic algorithms for bankruptcy predictions. *Expert Systems with Applications*, *11*(4), 407–413.

Bahrammirzaee, A. (2010). A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, *19*(8), 1165–1195.

Barron's 400. (2014, November 1st). Retrieved from `http://online.barrons.com/public/page/barrons_400.html`

Barry, C. B., & Winkler, R. L. (1976). Nonstationarity and portfolio choice. *Journal of Financial Quantitative Analysis*, *11*(02), 217–235.

Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal of Computing*, *6*(2), 154–160.

Bernardo, D., Hagras, H., & Tsang, E. (2013). A genetic type-2 fuzzy logic based system for financial applications modelling and prediction. In *IEEE international conference on fuzzy systems (FUZZ), 2013* (pp. 1–8).

Beyer, H.-G., & Sendhoff, B. (2007). Robust optimization–A comprehensive survey. *Computer methods in applied mechanics and engineering*, *196*(33), 3190–3218.

Brandt, M. W., & Santa-Clara, P. (2006). Dynamic portfolio selection by augmenting the asset space. *Journal of Finance*, *61*(5), 2187–2217.

Branke, J., & Schmidt, C. (2003). Selection in the presence of noise. In *Genetic evolutionary computation: GECCO 2003* (pp. 766–777).

Brennan, M. J. (1975). The optimal number of securities in a risky asset portfolio when there are fixed costs of transacting: Theory and some empirical results. *Journal of Financial Quantitative Analysis*, *10*(03), 483–496.

Çakmak, U., & Özekici, S. (2006). Portfolio optimization in stochastic markets. *Mathematical Methods of Operations Research*, *63*(1), 151–168.

Çanakoğlu, E., & Özekici, S. (2009). Portfolio selection in stochastic markets with exponential utility functions. *Annals of Operations Research*, *166*(1), 281–297.

Cantú-Paz, E. (2004). Adaptive sampling for noisy problems. In *Genetic evolutionary computation–GECCO 2004* (pp. 947–958).

Chang, Y.-H. (2010). Adopting co-evolution and constraint-satisfaction concept on genetic algorithms to solve supply chain network design problems. *Expert Systems with Applications*, *37*(10), 6919–6930.

Chen, A. H., Fabozzi, F. J., & Huang, D. (2012). Portfolio revision under mean-variance and mean-CVaR with transaction costs. *Review of Quantitative Finance and Accounting*, *39*(4), 509–526.

Chen, A. H., Jen, F. C., & Zionts, S. (1971). The optimal portfolio revision policy. *Journal of Business*, 51–61.

Chen, A. H., Jen, P. C., & Zionts, S. (1972). Portfolio models with stochastic cash demands. *Management Science*, *19*(3), 319–332.

Chen, A.-P., & Chang, Y.-H. (2005). Using extended classifier system to forecast S&P futures based on contrary sentiment indicators. In *The 2005 IEEE congress on evolutionary computation* (Vol. 3, pp. 2084–2090).

Chen, G., & Chen, X. (2011). A hybrid of adaptive genetic algorithm and pattern search for stock index optimized replicate. In *2nd international conference on artificial intelligence, management science and electronic commerce (AIMSEC), 2011* (pp. 4912–4915).

Chen, J.-S., Hou, J.-L., Wu, S.-M., & Chang-Chien, Y.-W. (2009). Constructing investment strategy portfolios by combination genetic algorithms. *Expert Systems with Applications*, *36*(2), 3824–3828.

Chen, S.-H. (2003). Agent-based computational macro-economics: A survey. In *Meeting the challenge of social problems via agent-based simulation* (pp. 141–170).

Chen, S.-H., & Lee, W.-C. (1997). Option pricing with genetic algorithms: A second report. In *International conference on neural networks, 1997* (Vol. 1, pp. 21–25).

Chen, Y., & Hirasawa, K. (2010). Generating trading rules on the stock markets with robust genetic network programming using variance of fitness values. In *Proceedings of SICE annual conference 2010* (pp. 3095–3102).

Chiu, M. C., & Li, D. (2006). Asset and liability management under a continuous-time mean–variance optimization framework. *Insurance: Mathematics and Economics*, *39*(3), 330–355.

Constantinides, G. M. (1979). Multiperiod consumption and investment behavior with convex transactions costs. *Management Science*, *25*(11), 1127–1137.

Corne, D. W., Knowles, J. D., & Oates, M. J. (2000). The Pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel problem solving from nature PPSN*

VI (pp. 839–848).

da Costa Moraes, M. B., & Nagano, M. S. (2014). Evolutionary models in cash management policies with multiple assets. *Economic Modelling*, *39*, 1–7.

de Araujo, R., Madeiro, F., de Sousa, R. P., Pessoa, L. F., & Ferreira, T. (2006). An evolutionary morphological approach for financial time series forecasting. In *IEEE congress on evolutionary computation (CEC), 2006* (pp. 2467–2474).

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (Vol. 16). John Wiley & Sons.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture notes in computer science*, *1917*, 849–858.

Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A. K., & Padmanabhan, D. (2009). Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *13*(5), 1054–1074.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002, Apr). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

de Brito, R. F., & Oliveira, A. L. (2012). Comparative study of FOREX trading systems built with SVR+GHSOM and genetic algorithms optimization of technical indicators. In *IEEE 24th international conference on tools with artificial intelligence (ICTAI), 2012* (Vol. 1, pp. 351–358).

DeJong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems* (Unpublished doctoral dissertation). Univ. Michigan.

del Arco-Calderón, C. L., Vinuela, P. I., & Castro, J. C. H. (2004). Forecasting time series by means of evolutionary algorithms. In *Parallel problem solving from nature-PPSN VIII* (pp. 1061–1070).

Desai, R., Lele, T., & Viens, F. (2003). A Monte-Carlo method for portfolio optimization under partially observed stochastic volatility. In *Computational intelligence for financial engineering, 2003. proc. 2003 ieee int. conf.* (pp. 257–263).

Diamond, W. (2001). *Practical experiment designs: for engineers and scientists*. John Wiley & Sons.

Diamond, W. (2006). *NIST/SEMATECH e-handbook of statistical methods.* Retrieved from `http://www.itl.nist.gov/div898/handbook`

Donate, J. P., & Cortez, P. (2014). Evolutionary optimization of sparsely connected and time-lagged neural networks for time series forecasting. *Applied Soft Computing*, *23*, 432–443.

Dumas, B., & Luciano, E. (1991). An exact solution to a dynamic portfolio choice problem under transactions costs. *Journal of Finance*, *46*(2), 577–595.

Elton, E. J., & Gruber, M. J. (1974a). The multi-period consumption investment problem and single period analysis. *Oxford Economic Papers*, 289–301.

Elton, E. J., & Gruber, M. J. (1974b). On the optimality of some multiperiod portfolio selection criteria. *Journal of Business*, 231–243.

Fagerland, M. W., & Sandvik, L. (2009). The Wilcoxon-Mann-Whitney test under scrutiny. *Statistics in medicine*, *28*(10), 1487.

Fama, E. (1970a). Multiperiod consumption-investment decisions. *American Economic*

Review, 163-174.

Fama, E. (1970b). Stochastic network programming for financial planning problems. *Journal Finance*, *25*.

Ferreira, C. (2001). Gene expression programming: A new adaptive algorithm for solving problems. *Complex Systems*, *13*(2), 87–129.

Finger, M., & Wasserman, R. (2004). Approximate and limited reasoning: Semantics, proof theory, expressivity and control. *Journal of Logic and Computation*, *14*(2), 179–204.

Finkelstein, M., & Whitley, R. (1981). Optimal strategies for repeated games. *Advances Appl. Probability*, 415–428.

Fitzpatrick, J. M., & Grefenstette, J. J. (1988). Genetic algorithms in noisy environments. *Machine Learning*, *3*(2-3), 101–120.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *ICGA* (Vol. 93, pp. 416–423).

Franke, R. (1998). Coevolution and stable adjustments in the cobweb model. *Journal of Evolutionary Economics*, *8*(4), 383–406.

FTSE. (2014, December). Retrieved from `http://www.ftse.com/products/home`

Gandomi, A. H., & Alavi, A. H. (2011). Multi-stage genetic programming: A new strategy to nonlinear system modeling. *Information Sciences*, *181*(23), 5227–5239.

García, S., Quintana, D., Galván, I. M., & Isasi, P. (2014). Multiobjective algorithms with resampling for portfolio optimization. *Computing and Informatics*, *32*(4), 777–796.

Garcia-Almanza, A. L., & Tsang, E. P. (2006). Forecasting stock prices using genetic programming and chance discovery. In *12th international conference on computing in economics and finance.*

Gaspar-Cunha, A., Recio, G., Costa, L., & Estébanez, C. (2014). Self-adaptive MOEA feature selection for classification of bankruptcy prediction data. *The Scientific World Journal*, *2014*.

Gerard, P., Stolzmann, W., & Sigaud, O. (2002). YACS: A new learning classifier system using anticipation. *Soft Computing*, *6*(3-4), 216–228.

Ghosh, P., & Chinthalapati, V. (2014). Financial time series forecasting using agent based models in equity and FX markets. In *6th computer science and electronic engineering conference (ceec), 2014* (pp. 97–102).

Giulioni, G., D'Orazio, P., Bucciarelli, E., & Silvestri, M. (2015). Building artificial economies: From aggregate data to experimental microstructure. A methodological survey. In *Advances in artificial economics* (pp. 69–78). Springer.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning* (Vol. 412). Addison-Wesley Reading Menlo Park.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, *1*, 69–93.

Gonçalves, J. F., & Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, *17*(5), 487–525.

Goonatilake, S., Campbell, J. A., & Ahmad, N. (1995). Genetic-fuzzy systems for financial decision making. In *Advances in fuzzy logic, neural networks and genetic algorithms* (pp. 202–223). Springer.

Grefenstette, J., Gopal, R., Rosmaita, B., & Van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. In *Proceedings of the first international conference on genetic algorithms and their applications* (pp. 160–168).

Gunawan, S., & Azarm, S. (2005). Multi-objective robust optimization using a sensitivity region concept. *Structural and Multidisciplinary Optimization*, *29*(1), 50–60.

Gupta, P., Mehlawat, M. K., & Mittal, G. (2012). Asset portfolio optimization using support vector machines and real-coded genetic algorithm. *Journal of Global Optimization*, *53*(2), 297–315.

Hamida, S. B., Abdelmalek, W., & Abid, F. (2014). Applying dynamic training-subset selection methods using genetic programming for forecasting implied volatility. *Computational Intelligence*. doi: 10.1002/coin.12057

Hamilton, J. D. (1994). *Time series analysis*. Princeton, MA: Princeton university press Princeton.

Harik, G. R., Lobo, F. G., & Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, *3*(4), 287–297.

Harik, G. R., Lobo, F. G., & Sastry, K. (2006). Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA). In *Scalable optimization via probabilistic modeling* (pp. 39–61). Springer.

Hibiki, N. (2006). Multi-period stochastic optimization models for dynamic asset allocation. *Journal of Banking and Finance*, *30*(2), 365–390.

Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, *42*(1), 228–234.

Hirabayashi, A., Aranha, C., & Iba, H. (2009). Optimization of the trading rule in foreign exchange using genetic algorithm. In *Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 1529–1536).

Hochreiter, R. (2014). An evolutionary optimization approach to risk parity portfolio selection. *arXiv preprint arXiv:1411.7494*.

Hochreiter, R. (2015). Computing trading strategies based on financial sentiment data using evolutionary optimization. *arXiv preprint arXiv:1504.02972*.

Hochreiter, R., & Wozabal, D. (2010). Evolutionary estimation of a coupled Markov chain credit risk model. In *Natural computing in computational finance* (pp. 31–44). Springer.

Holmes, J. H., Lanzi, P. L., Stolzmann, W., & Wilson, S. W. (2002). Learning classifier systems: New models, successful applications. *Information Processing Letters*, *82*(1), 23–30.

Holyoak, K. J., & Holland, J. H. (1989). *Induction: Processes of inference, learning, and discovery*. MIT press.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1993). Multiobjective optimization using the niched Pareto genetic algorithm. *IlliGAL report*(93005), 61801–2296.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence,1994* (pp. 82–87).

Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & De Carvalho, A. P. L. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *39*(2), 133–155.

Hu, Y., Feng, B., Zhang, X., Ngai, E., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications*, *42*(1), 212–222.

Huang, C.-F., Chang, C.-H., Kuo, L.-M., Lin, B.-H., Hsieh, T.-N., & Chang, B.-R. (2012). A genetic-search model for first-day returns using IPO fundamentals. In *International conference on machine learning and cybernetics (ICMLC), 2012* (Vol. 5, pp. 1662–1667).

Huang, C.-F., Hsu, C.-J., Chen, C.-C., Chang, B. R., & Li, C.-A. (2015). An intelligent model for pairs trading using genetic algorithms. *Computational Intelligence and Neuroscience*, *501*, 939606.

Huang, H., Pasquier, M., & Quek, C. (2009). Financial market trading system with a hierarchical coevolutionary fuzzy predictive model. *IEEE Transactions on Evolutionary Computation*, *13*(1), 56–70.

Hughes, E. J. (2000). Evolutionary algorithm with a novel insertion operator for optimising noisy functions. In *Proceedings of 2000 congress on evolutionary computation* (Vol. 1, pp. 790–797).

Investorpoint.com. investor information systems. (2015, February). Retrieved from `http://www.investorpoint.com/stock/mtlqq-MotorsLiquidationCompany/price-history/`

Jiang, Y., Xu, L., Wang, H., & Wang, H. (2009). Influencing factors for predicting financial performance based on genetic algorithms. *Systems Research and Behavioral Science*, *26*(6), 661–673.

Jin, Y., & Branke, J. (2005, Jun). Evolutionary optimization in uncertain environments: A survey. *IEEE Transactions on Evolutionary Computation*, *9*, 303–317.

Jin, Y., & Sendhoff, B. (2003). Trade-off between performance and robustness: an evolutionary multiobjective approach. In *Evolutionary multi-criterion optimization* (pp. 237–251).

Jing, L. (2010). Data modeling for searching abnormal noise in stock market based on genetic algorithm. In *2010 international symposium on computational intelligence and design* (Vol. 2, pp. 129–131).

Johnson, D. S., & McGeoch, L. A. (1997). The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, *1*, 215–310.

Jun, T., & Lei, H. (2012). Genetic optimization of BP neural network in the application of suspicious financial transactions pattern recognition. In *International conference on management of e-commerce and e-government (ICMeCG), 2012* (pp. 280–284).

Kamin, J. H. (1975). Optimal portfolio revision with a proportional transaction cost. *Management Science*, *21*(11), 1263–1271.

Kampouridis, M., Chen, S.-H., & Tsang, E. (2012). Microstructure dynamics and agent-based financial markets: Can dinosaurs return? *Advances in Complex Systems*, *15*(supp02).

Kanungo, R. P. (2004). Genetic algorithms: Genesis of stock evaluation. *Economics WPA Working Paper*(0404007).

Karatahansopoulos, A., Sermpinis, G., Laws, J., & Dunis, C. (2014). Modelling and trading the Greek stock market with gene expression and genetic programing algorithms. *Journal of Forecasting*, *33*(8), 596–610.

Kellerer, H., Mansini, R., & Speranza, M. G. (2000). Selecting portfolios with fixed costs and minimum transaction lots. *Annals of Operations Research*, *99*(1-4), 287–304.

Kim, K.-j., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with applications*, *19*(2), 125–132.

Kingdon, J., Taylor, J., & Mannion, C. (1997). *Intelligent systems and financial forecasting*. Springer-Verlag New York, Inc.

Knowles, J., & Corne, D. (1999). The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In *Proceedings of the 1999 congress on evolutionary computation (CEC)* (Vol. 1).

Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary computation*, *8*(2), 149–172.

Köppen, M., & Yoshida, K. (2007). Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *Evolutionary multi-criterion optimization* (pp. 727–741).

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection* (Vol. 1). MIT press.

Krink, T., & Paterlini, S. (2011). Multiobjective optimization using differential evolution for real-world portfolio optimization. *Computational Management Science*, *8*(1-2), 157–179.

Lahsasna, A., Ainon, R. N., & Teh, Y. W. (2010). Credit scoring models using soft computing methods: A survey. *International Arab Journal of Information Technology*, *7*(2), 115–123.

Lakemeyer, G. (1994). Limited reasoning in first-order knowledge bases. *Artificial Intelligence*, *71*(2), 213–255.

Laumanns, M., & Ocenasek, J. (2002). Bayesian optimization algorithms for multi-objective optimization. In *Parallel problem solving from nature PSN VII* (pp. 298–307). Springer.

Leon-Garcia, A. (1989). *Probability, statistics and random processes for electrical engineering* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Li, D., & Ng, W.-L. (2000). Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, *10*(3), 387–406.

Li, M., Azarm, S., & Aute, V. (2005). A multi-objective genetic algorithm for robust design optimization. In *Proceedings of the 7th annual conference on genetic and evolutionary computation* (pp. 771–778).

Li, Z., Yang, H., & Deng, X. (2007). Optimal dynamic portfolio selection with earnings-at-risk. *Journal of Optimization Theory and Applications*, *132*(3), 459–473.

Lim, D., Ong, Y.-S., & Lee, B.-S. (2005). Inverse multi-objective robust evolutionary design optimization in the presence of uncertainty. In *Proceedings of the 7th annual workshop on genetic and evolutionary computation* (pp. 55–62).

Lim, M., & Coggins, R. J. (2005). Optimal trade execution: An evolutionary approach. In *The 2005 IEEE congress on evolutionary computation* (Vol. 2, pp. 1045–1052).

Lin, F., Liang, D., Yeh, C.-C., & Huang, J.-C. (2014). Novel feature selection methods to financial distress prediction. *Expert Systems with Applications*, *41*(5), 2472–2483.

Lipinski, P. (2007). ECGA vs. BOA in discovering stock market trading experts. In *Proceedings of the 9th annual conference on genetic and evolutionary computation* (pp. 531–538).

Lipinski, P. (2012). Parallel evolutionary algorithms for stock market trading rule selection on many-core graphics processors. In *Natural computing in computational finance* (pp. 79–92). Springer.

Liu, T., Zhao, J., & Zhao, P. (2012). Portfolio problems based on jump-diffusion models. *Filomat*, *26*(3), 573–583.

Lo, A., Mamaysky, H., & Wang, J. (2000, August). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*, *55*, 1705-1765.

Lohpetch, D., & Corne, D. (2009). Discovering effective technical trading rules with genetic programming: Towards robustly outperforming buy-and-hold. In *World congress on nature & biologically inspired computing (NaBIC), 2009* (pp. 439–444).

Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, *24*, 757–772.

Ma, I., Wong, T., Sankar, T., & Siu, R. (2004). Forecasting the volatility of a financial index by wavelet transform and evolutionary algorithm. In *IEEE international conference on systems, man and cybernetics, 2004* (Vol. 6, pp. 5824–5829).

Mahfoud, S., & Mani, G. (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence*, *10*(6), 543–566.

Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work*. *Journal of Finance*, *25*(2), 383–417.

Markose, S., Tsang, E., Er, H., & Salhi, A. (2001). Evolutionary arbitrage for FTSE-100 index options and futures. In *Proceedings of the 2001 congress on evolutionary computation* (Vol. 1, pp. 275–282).

Markowitz, H. (1952, March). Portfolio selection. *Journal of Finance*, *7*.

Markowitz, H. M., Lacey, R., Plymen, J., Dempster, M., & Tompkins, R. (1994). The general mean-variance portfolio selection problem [and discussion]. *Philosophical Transactions: Physical Sciences and Engineering*, 543–549.

Martinez-Jaramillo, S., & Tsang, E. P. (2009). An heterogeneous, endogenous and coevolutionary GP-based financial market. *IEEE Transactions on Evolutionary Computation*, *13*(1), 33.

Matsui, K., & Sato, H. (2009). A comparison of genotype representations to acquire stock trading strategy using genetic algorithms. In *International conference on adaptive and intelligent systems (ICAIS), 2009* (pp. 129–134).

Matsui, K., & Sato, H. (2010). Neighborhood evaluation in acquiring stock trading strategy using genetic algorithms. In *International conference of soft computing and pattern recognition (SoCPaR), 2010* (pp. 369–372).

Matsumura, K., & Kakinoki, H. (2014). Portfolio strategy optimizing model for risk management utilizing evolutionary computation. *Electronics and Communications in Japan*, *97*(8), 45–62.

McDonald, J. (2015). *Handbook of biological statistics.* Retrieved from `http://www.biostathandbook.com/kruskalwallis.html`

Mendes, J. J. d. M., Gonçalves, J. F., & Resende, M. G. (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers & Operations Research*, *36*(1), 92–109.

Metodología del IPC. (2014, September 1st). Retrieved from `http://www.bmv.com`

.mx/
Meucci, A. (2010). Quant nugget 2: Linear vs. compounded returns–common pitfalls in portfolio management. *GARP Risk Professional*, 49–51.
Meyer, T. P., & Packard, N. H. (1992). Local forecasting of high-dimensional chaotic dynamics. In *Santa Fe institute studies in the sciences of complexity-proceedings volume-* (Vol. 12, pp. 249–249).
Miller, J. F., & Thomson, P. (2000). Cartesian genetic programming. In *Genetic programming* (pp. 121–132). Springer.
Mulvey, J. M., & Vladimirou, H. (1992). Stochastic network programming for financial planning problems. *Management Science*, *38*(11), 1642–1664.
NASDAQ global indexes. (2015, February). Retrieved from `https://indexes.nasdaqomx.com/Index/Overview/NDX`
Ngai, E., Hu, Y., Wong, Y., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559–569.
Nguyen, H. D., Yoshihara, I., Yamamori, K., & Yasunaga, M. (2007). Implementation of an effective hybrid GA for large-scale traveling salesman problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *37*(1), 92–99.
Nikkei indexes. (2015, February 2nd). Retrieved from `http://indexes.nikkei.co.jp/en/nkave`
Nikolaos, L., & Iordanis, E. (2010). Default prediction and bankruptcy hazard analysis into recurrent neuro-genetic-hybrid networks to adaboost M1 regression and logistic regression models in finance. In *7th WSEAS international conference on engineering education* (pp. 22–24).
Notas sobre Índices. (2015, February). Retrieved from `http://www.bmv.com.mx/wb3/wb/BMV/BMV_notas_sobre_indices`
Ong, Y.-S., Nair, P. B., & Lum, K. (2006). Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Transactions on Evolutionary Computation*, *10*(4), 392–404.
Packard, N. H. (1990). A genetic learning algorithm for the analysis of complex data. *Complex Systems*, *4*(5), 543–572.
Paenke, I., Branke, J., & Jin, Y. (2006). Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Transactions on Evolutionary Computation*, *10*(4), 405–420.
Parracho, P., Neves, R., & Horta, N. (2011). Trading with optimized uptrend and downtrend pattern templates using a genetic algorithm kernel. In *IEEE congress on evolutionary computation (CEC), 2011* (pp. 1895–1901).
Patel, N. R., & Subrahmanyam, M. G. (1982). A simple algorithm for optimal portfolio selection with fixed transaction costs. *Management Science*, *28*(3), 303–314.
Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (2000a). Bayesian optimization algorithm, population sizing, and time to convergence. In *GECCO* (pp. 275–282).
Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (2000b). Hierarchical problem solving and the bayesian optimization algorithm. In *GECCO* (pp. 267–274).
Pérez Rave, J. I., & Jaramillo Álvarez, G. P. (2013). Relevant literary space on traveling salesman problem (TSP): Contents, classification, methods and fields of inspiration.

Produção, *23*(4), 866–876.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

Polanski, A. (2011). Genetic algorithm search for predictive patterns in multidimensional time series. *Complex Systems*, *19*(3), 195.

Ponsich, A., Jaimes, A. L., & Coello, C. A. C. (2013). A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, *17*(3), 321–344.

Potter, M. A., & De Jong, K. A. (1994). A cooperative coevolutionary approach to function optimization. In *Parallel problem solving from nature PSN III* (pp. 249–257). Springer.

Protopapas, M. K., Battaglia, F., & Kosmatopoulos, E. B. (2010). Coevolutionary genetic algorithms for establishing Nash equilibrium in symmetric cournot games. *Advances in Decision Sciences*, *2010*.

Radeerom, M. (2014). Automatic trading system based on genetic algorithm and technical analysis for stock index. *Change*, *1*(100), 100.

Ranković, V., Drenovak, M., Stojanović, B., Kalinić, Z., & Arsovski, Z. (2014). The mean-value at risk static portfolio optimization using genetic algorithm. *Computer Science and Information Systems*, *11*(1), 89–109.

Ravi, V., Kurniawan, H., Thai, P. N. K., & Kumar, P. R. (2008). Soft computing system for bank performance prediction. *Applied Soft Computing*, *8*(1), 305–315.

Rimcharoen, S., Sutivong, D., & Chongstitvatana, P. (2005). Prediction of the stock exchange of Thailand using adaptive evolution strategies. In *17th IEEE international conference on tools with artificial intelligence (ICTAI), 2005* (pp. p5–pp236).

Russel, S., & Norvig, P. (2010). *Artificial intelligence : A modern approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Ryan, C., Collins, J., & Neill, M. O. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In *Genetic programming* (pp. 83–96). Springer.

Safarzyńska, K., & van den Bergh, J. C. (2010). Evolutionary models in economics: A survey of methods and building blocks. *Journal of Evolutionary Economics*, *20*(3), 329–373.

Sarijaloo, A., & Moradbakloo, A. (2014). Asset management using genetic algorithm: Evidence from Tehran stock exchange. *Management Science Letters*, *4*(2), 221–226.

Sastry, K., & Goldberg, D. E. (2003). Scalability of selectorecombinative genetic algorithms for problems with tight linkage. In *Genetic and evolutionary computation (GECCO) 2003* (pp. 1332–1344).

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international conference on genetic algorithms* (pp. 93–100).

Schmidbauer, H., Rösch, A., Sezer, T., & Tunalioğlu, V. S. (2014). Robust trading rule selection and forecasting accuracy. *Journal of Systems Science and Complexity*, *27*(1), 169–180.

Shao, M., Smonou, D., Kampouridis, M., & Tsang, E. (2014). Guided fast local search for speeding up a financial forecasting algorithm. In *IEEE conference on computational intelligence for financial engineering & economics (CIFEr), 2014* (pp. 325–332).

Sharpe, W. F., Alexander, G. J., & Bailey, J. V. (1999). *Investments*. New Jersey, NJ: Prentice

Hall.

Shim, V. A., Tan, K. C., Chia, J. Y., & Al Mamun, A. (2013). Multi-objective optimization with estimation of distribution algorithm in a noisy environment. *Evolutionary Computation*, *21*(1), 149–177.

Sinha, A., Malo, P., Frantsev, A., & Deb, K. (2014). Finding optimal strategies in a multi-period multi-leader–follower Stackelberg game using an evolutionary algorithm. *Computers & Operations Research*, *41*, 374–385.

S&P Dow Jones indices. (2014, November 1st). Retrieved from `http://www.djaverages.com`

Steinbach, M. C. (2001). Markowitz revisited: Mean-variance models in financial portfolio analysis. *SIAM review*, *43*(1), 31–85.

Stolzmann, W. (2000). An introduction to anticipatory classifier systems. In *Learning classifier systems* (pp. 175–194). Springer.

Tapia, M. G. C., & Coello, C. A. C. (2007). Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *IEEE congress on evolutionary computation* (Vol. 7, pp. 532–539).

Teknomo, K. (2006). *Recursive average and variance.* Retrieved from `http://people.revoledu.com/kardi/tutorial/RecursiveStatistic/index.html`

Tezuka, M., Munetomo, M., & Akama, K. (2007). Genetic algorithm to optimize fitness function with sampling error and its application to financial optimization problem. In *Evolutionary computation in dynamic and uncertain environments* (pp. 417–434). Springer.

Trautmann, H., Mehnen, J., & Naujoks, B. (2009). Pareto-dominance in noisy environments. In *IEEE congress on evolutionary computation (CEC), 2009* (pp. 3119–3126).

Tsang, E., Markose, S., Garcia, A., Almad Garcia, & Er, H. (2006). EDDIE for discovering arbitrage opportunities. In *International conference on numerical methods for finance, 2006.*

Tsang, E., Yung, P., & Li, J. (2004). EDDIE-automation, a decision support tool for financial forecasting. *Decision Support Systems*, *37*(4), 559–565.

Tsang, E. P., Li, J., Markose, S., Er, H., Salhi, A., & Iori, G. (2000). EDDIE in financial decision making. *Journal of Management and Economics*, *4*(4).

Tsang, E. P., & Martinez-Jaramillo, S. (2004). Computational finance. *IEEE Computational Intelligence Society Newsletter*, *3*(8).

Valenzuela-Rendón, M. (1991). The fuzzy classifier system: A classifier system for continuously varying variables. In *Proceedings of the fourth international conference on genetic algorithms pp346-353, Morgan Kaufmann I* (Vol. 991, pp. 223–230).

Valenzuela-Rendón, M. (2003). The virtual gene genetic algorithm. In *Genetic and evolutionary computation GECCO 2003* (pp. 1457–1468).

Valenzuela-Rendón, M., & Uresti-Charre, E. (1997). A non-generational genetic algorithm for multiobjective optimization. In *in proceedings of the seventh international conference on genetic algorithms.*

Vanstone, B., & Tan, C. (2003). A survey of the application of soft computing to investment and financial trading. *Information Technology Papers*, 13.

Varetto, F. (1998). Genetic algorithms applications in the analysis of insolvency risk. *Journal of Banking & Finance*, *22*(10), 1421–1439.

Verikas, A., Kalsyte, Z., Bacauskiene, M., & Gelzinis, A. (2010). Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey. *Soft Computing*, *14*(9), 995–1010.

Wagman, L. (2003). Stock portfolio evaluation: An application of genetic-programming-based technical analysis. *Genetic Algorithms and Genetic Programming at Stanford*, *2003*, 213–220.

Wagner, N., Michalewicz, Z., Khouja, M., & McGregor, R. R. (2007). Time series forecasting for dynamic environments: the DyFor genetic program model. *IEEE Transactions on Evolutionary Computation*, *11*(4), 433–452.

Wang, W., Hu, J., & Dong, N. (2014). A convex-risk-measure based model and genetic algorithm for portfolio selection. *Mathematical Problems in Engineering*, *2014*.

WFE monthly reports. (2015, December). Retrieved from `http://www.world-exchanges.org`

Whitley, L. D. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *ICGA* (Vol. 89, pp. 116–123).

Wiesmann, D., Hammel, U., & Back, T. (1998, Jul). Robust design of multilayer optical coatings by means of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *2*, 162–167.

Wilson, S. W. (1994). Zcs: A zeroth level classifier system. *Evolutionary computation*, *2*(1), 1–18.

Xia, H., Zhuang, J., & Yu, D. (2014). Multi-objective unsupervised feature selection algorithm utilizing redundancy measure and negative epsilon-dominance for fault diagnosis. *Neurocomputing*, *146*, 113–124.

Xiong, H., Xu, Y., & Xiao, Y. (2009). Comparative analysis of multi-period portfolio strategies. In *Business intelligence and financial engineering, bife'09. int. conf.* (pp. 266–269).

Yahoo! finance. (2015, February). Retrieved from `http://finance.yahoo.com/`

Yaman, A., Lucci, S., & Gertner, I. (2014). Evolutionary algorithm based approach for modeling autonomously trading agents. *Intelligent Information Management*, *2014*.

Yi, L. (2010). Multi-period portfolio selection with transaction costs. In *2nd IEEE international conference on information and financial engineering (ICIFE), 2010* (pp. 98–103).

Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L., & Thiele, L. (2001). *SPEA2: Improving the strength Pareto evolutionary algorithm.* Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK).

Vita

Rubén Anton Aguilar Rivera was born in Chihuahua, México, on July 17th, 1981. He earned the Bachelor of Science in Electronics and Communications Engineering degree from ITESM, Campus Monterrey, on December 2003. Also, he earned the Master of Sciences in Automation degree from ITESM, Campus Monterrey, on May 2006. He worked as an assistant and laboratory instructor at that time. After some years working in the steel industry, he was accepted in the Doctorate of Information and Communications Technologies program at ITESM, Campus Monterrey, on August 2011. He earned the Doctor of Philosophy in Information Technologies and Communications degree, major in Intelligent Systems, on December 2015. Let's see what happens next.

This doctoral dissertation was typed in using LaTeX 2_ε[a] by Rubén Anton Aguilar Rivera.

[a]The style file `phdThesisFormat.sty` used to set up this dissertation was prepared by the Center of Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus