Evolving Decision Models for Asset Selection in Equity Portfolio Management

Adam Mostafa Ghandar

Submitted in fulfillment of the requirements
of the degree of Doctor of Philosophy

Supervisor: Professor Dr. Zbigniew Michalewicz
Supervisor: Professor Ralf Zurbrugg
Supervisor: Dr. Martin Schmidt

October 2009

School of Computer Science
The University of Adelaide

For my parents and brother

Contents

Abstract

This thesis contributes an approach to equity portfolio management using computational intelligence methodologies. The focus is on generating an automated financial reasoning, with a basis in financial research, through searching a space of semantically meaningful propositions. The objective function to compare propositions is defined by a trading simulation.

In comparison with classical financial modeling, this approach allows continual adaptation to changing market conditions and a non-linear solution representation. Compared with other computational intelligence approaches, the focus is on a holistic design that integrates financial research with machine learning.

A major aim of the thesis is to develop methodologies for learning investment decision models for portfolio management that can adapt with market processes, the applications performance and the environment. It is toward this goal that we make use of a cross-disciplinary approach that combines an evolving fuzzy system with financial theory to perform key procedures at the conceptual level (as opposed to the execution of trades, storing information, etc.) We evaluate the methods developed in out of sample trading over historic data. The testing is designed to be realistic, for instance considering factors such as transaction costs, stock mergers and data snooping issues. We test scenarios for European and Australian stock markets in different economic conditions. It is found that the methodology is able to outperform the market in these cases.

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

16 October 2009

Acknowledgments

It has been a truly great experience to have had the opportunity to work with Professor Dr. Zbigniew Michalewicz, an incredibly impressive person without whom this thesis would never have reached this stage. It was also a great experience to work with Professor Ralf Zurbruegg, whose depth of understanding of financial research was invaluable. I also acknowledge the excellent advice and help of Dr. Martin Schmidt who it is also a pleasure to have met. In addition, I would like to thank Thuy Duong-To for her help and advice about using financial analysis techniques, and Frank Neumann and Jens Kroeske for their collaboration and help with mathematical analysis. I would also like to thank the School of Computer Science and Finance and all of the staff for helping make the school a great environment, notably Tracey Young for help and advice on arrangements and procedures associated with my candidature. I also thank Kate Deller-Evans for her assistance in correcting the final draft.

Chapter 1

Introduction

During the last two decades there has been a trend of increasing use of computer assisted modeling of financial processes and environments by practitioners and academics in finance. The growth of “quant” funds (managed funds for which investment decisions are based on quantitative analysis) has highlighted the interest of the financial community in examining computational methods to assist in portfolio management. Over the same period, the field of Computational Intelligence has developed and matured but has not as yet been extensively applied in this area, although this is changing as these methods become better known. Computational Intelligence is an area that has as its goal the creation of intelligence in ways that do not necessarily mimic the processes of human intelligence, but instead take advantage of the characteristics of computation to create intelligent systems. The field includes the techniques of evolutionary computation, fuzzy systems and neural networks.

This thesis develops ways for applying evolutionary computation and fuzzy systems for portfolio management. Some benefits or contributions of the approach include: the development of prediction models that guide investment decisions with a high fidelity in capturing market processes by analyzing large quantities of potentially relevant information and interpreting the resulting knowledge; and the ability to continually adapt these models as market conditions change. In this way we will develop an integrated system that automates fundamental aspects of financial decision making. This will provide a structured and comprehensive assistance to a complex decision process.

It is also a goal of this thesis that recommendations from the models should be verifiable in that decision guidance is able to be understood and checked by human analysts or stored to explain earlier actions. The importance of an approach that enables verification of recommendations is important, particularly in the light of recent information regarding the global financial crisis and poor performance of many investment funds. A "black box" style system, a classic example being a neural network, could have difficulty in gaining the acceptance of the fund management industry because of issues such decision justification or an inability to check a model. Fuzzy systems can facilitate structured rule base models expressible in natural language. Decision models that can be understood by human analysts are also amenable to being supplemented by user defined guidance.

We use the Adaptive Business Intelligence methodology in our approach. Adaptive Business Intelligence (ABI) provides a framework that combines elements of predictive modeling, forecasting, optimization, and adaptability for solving real world problems [78]. This is facilitated by Computational Intelligence techniques. The traditional approach to the design of business intelligence information systems involves processing data to obtain information, and then using statistical data analysis techniques, to infer knowledge from the information. For example, it might be the case a company would find that 60% of its customers are between, say, 18 and 30 years old. The knowledge about the business and operations is reported to end users who are responsible for using the knowledge as a basis for action (e.g. decisions in formulating a marketing plan). A further logical step is made in ABI to interpret and apply this knowledge. This leads to a new generation of intelligent business information systems which recommend courses of action or even implement decisions. The intelligent component adds another type of functionality in which knowledge is applied. Intelligent decision support requires adaptation as circumstances change, prediction to anticipate the future, and optimization to find the best possible decisions with respect to objectives. ABI has been used in the development of many real world systems real world intelligent decision support systems, for example see [4, 79, 81]. We use the principles of ABI in a way that also incorporates financial application knowledge.

Some examples of how application domain knowledge is incorporated into the basic structure of the design and approach to portfolio management that is developed in this thesis include the following: the specification of the problem as as a stock selection and valuation issue rather than a traditional time series forecasting problem; by the handling of input information set and forecasting model components to develop an approach for asset valuation similar to that used in practice by financial analysts; and by considering *risk adjusted performance*.

With reference to these goals, ideas and methodologies we develop adaptive computational intelligence methods for financial portfolio management. Adaptation is a key part of intelligence that is essential when operating in a dynamic environment. Therefore, the primary aim of this research may be stated clearly as: *to construct an adaptive decision support system to intelligently manage a portfolio of assets over time*. In addition, a number of auxiliary aims include the following:

- to develop models to represent relative asset valuation problems (the relative value of one company stock over another in terms of future performance),
- to convert financial models and information sets into entities able to be processed by intelligent information systems for instance representations and objective definitions,
- to develop of techniques addressing adaptation: specifically handling the time varying and adaptive aspects to effect changes to the prediction and optimisation components. Implementing these more advanced methods (which are also not specifically applied in the literature to the problems we are considering in finance) will add substantial value to the work,
- to apply the developed technology within a financial portfolio management context and measure its performance gains in a meaningful way with respect to the field of application,
- to develop application frameworks and prototypes that demonstrate and which would enable the technology to be applied in practice,
- to provide insight into financial research questions.

The addressing of financial research questions will occur in several ways that involve analysis of the performance of the systems that implement a computational approach in a financially sound way. Out of necessity testing of the approaches developed makes use of simulation and historic data. It is fundamental that this analysis involves out-of-sample testing on data systems have not "seen" previously. Furthermore, these are designed to be realistic by considering aspects such as transaction costs and data snooping, for example survivorship bias where a data set only includes stocks that exist at the end of a test period.

Also on the topic of the contribution to financial research, there is at present a reemergence of interest in determining the profitability of utilizing trading rules within financial academia, including technical indicators that are included as inputs to the adapting models developed in this thesis. Although the application of examining historical price and volume relationships to predict future price movements have never left the forefront of the investment community, academics have generally been more wary of the benefits that can be derived from such an exercise. One primary reason is that it would refute the theory of market efficiency where, at the very least, all past information that is useful in valuing an asset today should already be reflected in the price. If this were not the case then it would put into question the validity of numerous asset pricing models that are based on markets being efficient, particularly for there to be a measured tradeoff between the underlying risk structure of the stock and its expected return.

During the course of writing this thesis a number of articles have been published, these include:

1. Ghandar, A., Michalewicz, Z., Schmidt, M., To, T.-D., and Zurbruegg, R., Computational Intelligence for Evolving Trading Rules, *IEEE Transactions on Evolutionary Computation*, Vol.13, No.1, 2009, pp. 71–86.

2. Ghandar, A., Michalewicz, Z., and Zurbruegg, R., Return Performance Volatility and Adaptation in an Automated Technical Analysis Approach to Portfolio Management, *Journal of Intelligent Systems in Accounting Finance and Management*, Vol.16, No.1, 2009, pp.127–146.

3. Ghandar, A., Michalewicz, Z., Schmidt, M., To, T.-D., Zurbruegg, R., Evolving Trading Rules, *Chapter 10, Success in Evolutionary Computation*, Yang, A., Shan, Y., and Bui, T. (Editors), Springer, 2008.

4. Ghandar, A., Michalewicz, Z., Zurbruegg, R., Learning Multi-Criteria Fuzzy Rule Based Decision Models for Hedge Fund Management, *Meta Heuristics International Conference*, Hamburg, Germany, July 13–16, 2009

5. Buckley, M., Ghandar, A. Michalewicz, Z. and Zurbruegg, R., Evaluation of Intelligent Quantitative Hedge Fund Management, *Proceedings IEEE Conf. on Evol. Computation (CEC2009).*

6. Ghandar, A., Michalewicz, Z., To, T.-D., and Zurbruegg, R., The Performance of an Adaptive Portfolio Management System, *Proceedings of the 2008 Congress on Evolutionary Computation (CEC2008)*, Hong Kong, June 1 6, 2008.

7. Kroeske, J., Ghandar, A., Michalewicz, Z., and Neumann, F., Learning Fuzzy Rules with Evolutionary Algorithms-an Analytic Approach, *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature (PPSN 2008)*, Dortmund, September 13 – 17, 2008.

8. Ghander, A.. Michalewicz, Z., Schmidt, M., To, T., Zurbruegg, R., A Computational Intelligence Portfolio Construction System for Equity Market Trading,*Proceedings of the 2007 Congress on Evolutionary Computation (CEC2007)*, Singapore, September 26 – 29, 2007.

9. Ghander, A.. Michalewicz, Z., Schmidt, M., To, T., Zurbruegg, R., Computational Intelligence for Evolving Trading Rules, *Proceedings of the 20th Annual Australasian Finance and Banking Conference*, Sydney, December 12 – 14, 2007.

The thesis is organized as follows. Chapter 2 provides an introduction to the topic of portfolio management and the tools and methodology to approach important problems in this area. Chapter 3 surveys previous work in this field. Chapter 4 provides the design of a particular computational intelligent portfolio management system; and an analysis of the performance is provided in Chapter 5 together with discussion of its significance from a financial academic perspective. Chapter 6 discusses the development of a number of mechanisms to incorporate adaptive intelligence. Chapter 7 contains a theoretical analysis of the representation and development of the optimization and prediction aspects of the design of solutions to the problem of stock price forecasting using many explanatory variables. Chapter 8 incorporates consideration of different objectives and greatly extends the input data set of explanatory variables to include balance sheet, accounting and macro economic data. Chapter 9 provides an exposition of an application including user interface and design aspects. Finally, we draw conclusions and provide direction for future work in Chapter 10.

Chapter 2

Objectives and Problem Description

In this chapter the objectives and task specifications of the problem domain (portfolio management) are discussed, followed by background information about the computational methods of approach. We give a context to the ideas and applications developed later in the thesis. Melding domain knowledge with intelligent adapting computation techniques are unique components of the asset valuation and investment approach developed.

2.1 Portfolio Selection and Management

Modern portfolio theory, extrapolated by Markowitz (see [74, 73]), decomposes volatility into *systematic risk* and *unsystematic risk*. The systematic risk component reflects how changes in market conditions affect portfolio values. The unsystematic risk component is unique for each portfolio. By enforcing constraints on portfolio structure and contents it is possible to reduce the unsystematic risk component significantly so that the main source of risk is systematic, which enables general application of some basic principles for risk management. This is achieved mainly by ensuring a number of different stocks from several industry sectors, termed *diversification*. A well diversified portfolio should have return that compensates for the systematic risk component. In this way the return may be managed with respect to risk by using mathematical models.

Let us define these concepts more formally beginning with some notation. Denote $r_{m,t}$ the return of a market m – defined as the index return – on a day t (or time instance); let $r_{p,t}$ be the return achieved by a portfolio, p, constructed from stocks in m at t; and finally denote the risk free interest rate at t by $r_{\mathrm{f},t}$. The portfolio and the market return on any day t are defined: $r_t = ln\,(p_t/p_{t-1})\,,$ where p_t is the price or index value at t. One way of relating risk and return is using the *Capital Asset Pricing Model* (CAPM) model. The systematic risk β_p of portfolio p with return $r_{p,t}$ is as follows:

$$r_{p,t} - r_{\mathrm{f},t} = \alpha_p + \beta_p(r_{m,t} - r_{\mathrm{f},t}) + e. \tag{2.1}$$

The excess return over the risk free rate, $r_{p,t} - r_{\mathrm{f},t}$, of any portfolio should be fully explained by its level of systematic risk β_p and the market risk premium $r_{m,t} - r_{\mathrm{f},t}$. In an efficient market the alpha value for portfolio returns, α_p, should be zero because it would not be possible for traders to make a profit from past data as all relevant information for pricing a security today would be incorporated in today's price. The term β_p represents risk and explains the difference in returns by additional risk inherent in the portfolio p over the market. A positive alpha of a portfolio (or asset) can be explained predominantly by one of two possible reasons: good stock picking ability of the portfolio manager or exposure to unaccounted risk factors beyond the scope of the CAPM model. Patterns in average returns that are not explained by the standard CAPM are termed *anomalies*.

An important issue in the development of financial portfolio theory has been the investigation of empirical evidence of the existence of these anomalies in the returns of some common stocks, significantly in [33]. The possibility that a number of additional factors relate to stock prices and underlying companies explain excess returns above the market index. The occurrence of pricing anomalies is greatly reduced when only two additional factors are considered as in the three-factor Fama and French asset pricing model which extends the CAPM to include company size (total market value) and price to book value in addition to the market index [33] . The price to book value ratio is the ratio of the market value of a stock to the book value.

A generalizable multi-factor alpha regression model that relates return to several risk factor premiums $\beta_1, \ldots, \beta_k$ is able to be defined precisely. There are k factors, each responsible for a portion portfolio of the returnf_k such that:

$$r_{p,t} = \alpha_p + \beta_1 f_1 + \beta_2 f_2 + \ldots + \beta_k f_k + e, \tag{2.2}$$

e is an error term. A four-factor model where price momentum (change in price over previous four months) is the fourth factor in addition to the three-factor Fama and

French model is a standard used in industry and academia. As a tool to understand portfolio dynamics all additional returns or positive alpha values can be explained in terms of unconsidered risk factors. Then factors can be added to the regression model to achieve a better fit and by assumption better explain returns.

In cases where markets are not efficient and participants actions are not always rational, other explanations for anomalies are considered. In this case, the dynamics of the group behavior of market participants would be factors. If the pricing of listed market items do not accurately reflect the risk premium because of irrational pricing tendencies of market participants, then corrections would take place leading to excess returns being observed from time to time. For example, securities could become *undersold* or *oversold* by participants so that prices become unreasonably high or low. Such events could be discovered by analyzing the time series of stock prices in a process termed *technical analysis*.

2.2 Financial Modeling

Financial thinking has evolved during recent decades with a shift away from absolute faith in market efficiency to the position that markets are only "almost" efficient and behavioral explanations are required to account for exceptions. In an efficient market the CAPM alpha should always be zero. The implication of the changing consensus view of the market is to imply that the best strategy is not necessarily to attempt to *passively* attain returns that follow the market index. Instead *active* stock picking approaches are used to attempt to attain return on investment in excess of the market.

One of the main reasons for this shift has been an increasing body of empirical results that contradict the hypothesis that the prices of stocks and other market instruments are, for the purposes of prediction, random. As a consequence *behavioral models* are used to explain some pricing effects. Some examples of patterns found in stock prices used to obtain returns in excess of the market over long periods include the profitability of momentum strategies [53]. Other technical indicator strategies such as Bollinger bands, moving averages and relative strength index [16] also have been shown to promote excess risk adjusted returns. Another category is cyclical trends, for example the "January effect" [26] where the previous year's underperforming stocks outperform in the following January because investors and managed funds sell off of underperforming assets in January.

To some extent, the growing body of empirical evidence cited from academic research above is a result of developments in information technology. In fact in recent decades, computers have had a very large impact on operations at all levels in the financial sector. Advances in computing power and availability encourage application of complex mathematical models and statistical methods that may be readily applied to large volumes of data in electronic format. The culmination of this influence on portfolio management is in the rapidly expanding field of Quantitative Investment (QI). Applied to portfolio management QI is defined flexibly as "an approach to portfolio management that takes full advantage of today's better understanding of the market and greater technological capacity for sophisticated investing" [70]. QI involves utilization of these ideas and techniques for three main activities: return forecasting, portfolio construction and optimization, and performance measurement of resulting portfolios (Figure 2.1). There is a clear feedback loop as the performance of managed portfolios over time logically should cause the model and portfolio construction methods to be either maintained or adjusted. Although it nevertheless remains an open problem to adapt quantitative trading models as quickly as a traditional analyst because of reliance on performance analysis and historical data.

Portfolio building, implicitly or explicitly, involves a valuation based on forecasting future asset prices from a basis of current knowledge. In order to effectively understand and adjust a model it is necessary to analyze the performance of portfolios managed using the forecasting model and resulting constructed portfolios over time (Fig. 2.1). A (conceptual) multi-factor model relating risk and return with a time component is expressed as follows:

$$r_{i,t} = \alpha_i + \beta_{i,1} f_{1,t} + \beta_{i,2} f_{2,t} + \ldots + \beta_{i,k} + e_{i,t}, \tag{2.3}$$

where $r_{i,t}$ is the return of a stock i at time t, $f_{1,t}, \ldots, f_{k,t}$ are k returns due to factors, $\beta_{i,1}, \ldots, \beta_{i,k}$, $f_{k,t}$ are multipliers for the risk of including facts and $e_{i,t}$ is an error term [70]. This expression is a prototype for a prediction model that relates return to risk (by the β terms) and is also divided into model factors. it also is the case that the terms $f_{1,t}, \ldots, f_{k,t}$ can change over time to model a changing impact of factors over time. By using fuzzy rule bases in this work the model specification is not linear and doesn't use linear regression to find solutions. And, in addition the selection of the period of data for developing the model can change. We apply computational intelligence to dynamically build a multi-factor price forecasting model that anticipates and adapts the weight, possibly zero, of factors over time. The paridigm of computational intelligence is distinct from philosophies of artificial intelligence that attempt to precisly imitate human reasoning in that it involves harnessing the unique abilities of computers to produce "intelligence".

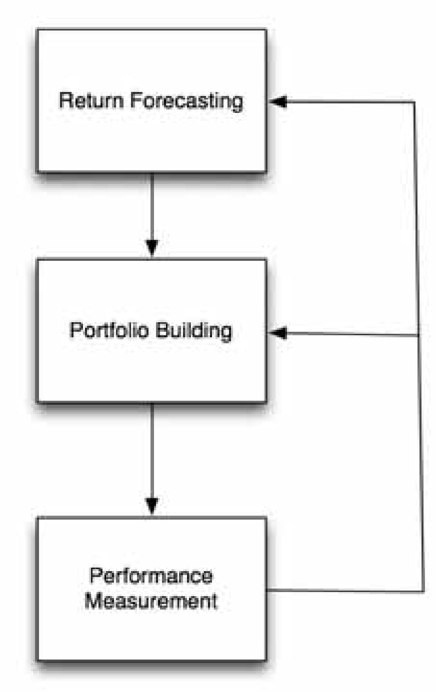


Figure 2.1: The three main processes of Quantitative Equity Portfolio Management and their relation. The portfolio optimization process takes, as at least one input, information from the forecasting model. The effectiveness of the forecasting model and portfolio construction methodology can be gauged by performance measurement and includes features such as consistency of returns over time, comparison to benchmarks and so on as well as standards such as the annual rate of return.

2.3 Information Set

Now that we have presented some aspects of the forecasting model we discuss the model factors, $f_{1,t}, \ldots, f_{k,t}$, in equation 2.3. There are at least three distinct classifications of information that have been used to explain returns:

- Market and macro economic indicators
- Fundamental indicators
- Technical indicators

Macro economic indicators, such as a country's gross domestic product, and interest rates; and other variables like currency exchange rates and commodity prices, have a significant impact on equity markets. Fundamental analysis is a natural approach involving consideration of the assets underlying market securities, for instance the companies whose stock is listed on the stock exchange. Using sources, such as accounting data and even natural language data such as news, it is accepted that it is possible to identify assets that provide good value. Some important criteria include cash flow, total company earnings, derivative information such as the ratio of earnings to share price and others. Analysts give different importance to these factors depending on industry groups or sectors, market conditions, economic conditions and even personal experience. For example importance may be given to earnings before tax and other liabilities to give an indication of the underlying strength of a company's position, and there are many reasons for variations, if for instance a firm operates in an industry that is highly regulated and subject to many taxes it may be the case that important aspects of its position relative to companies in other industries are hidden. In academia the explanatory performance of potential models is often compared with a standard such as the four-factor model discussed in the previous subsection.

Technical analysis is widely used in practice. It involves constructing and applying technical analyses of price and volume movements. This approach can extract information about market expectations, particularly behavioral effects. These indicators are divided into the following categories by their use in modeling different types of price movements: moving average, momentum, oscillation, and breakout indicators and also indicators based on volume, or price and volume rather than only price. Moving averages are often used to identify trends and to smooth out fluctuations due to daily or short, unsustained changes, depending on the period to calculate the average. New trends are identified when a moving average series crosses the price, or a shorter period average crosses a longer average. Oscillating indicators are used to identify cyclic patterns in price movements by compressing observations into a range, possibly giving more weight to recent points, and then generating buy or sell signals appropriately when extremes in the range are reached. Breakout indicators, as suggested by their name, are designed to catch significant changes in price direction at an early stage, for example a movement well outside the standard deviation of the mean historic returns is an indication that an unusual trend is emerging as opposed to a cyclic occurrence. Volume data is an important input component and an indicator of market sentiment with links to behavioral aspects of market activity. In general a market is considered strong by technical analysts if price and volume are both increasing.

From a financial view point, the model inputs are the kernel of the methodology. This is because, from a high level, the existence of relationships between these factors and returns are the model definition. We assume and implement a methodology for equity markets. However, the principles can be applied to other listed market items such as options, warrants, and other investment instruments or markets in general.

From a top down viewpoint the inputs are divided into macro economic factors which operate at a global level such as gross domestic product (GDP), the value of the whole market and also information related to sectors of the economy in which individual companies conduct their business such as the resources sector, consumer discretionary sector etc. Fundamental company information also exists outside the market but is specific to each stock, this information includes information about cash flows, earnings an so on. A separate category is data from trading activity within the market and includes things such as price and volume series and market capitalization. The raw inputs are given in Table 2.3. These raw inputs are processed to produce different classes of information as given in Figure 2.2.

The model factors used in this thesis are constructed from these data types. Processed data, a structure for the universe of models is imposed in a way that influences the model and the optimization process to use some predefined information classes. Other techniques such as genetic programming [52] have been used to optimize the factors and find equations directly from raw data. When flexible model specification or learning is used there is an implied balance on the one hand between providing no pre-processing and on the other defining the factors restrictively so as to prevent exploration. Figure 2.2 is a matrix of the information sets and analysis methods that are used and provides a model of how technical, fundamental and macro economic data is considered in financial analysis styles. The number of factors is termed the breadth of the model. In the Figure 2.2 the following acronyms are used: MA stands for moving average, PM to price momentum, Sector to industry sectors (e.g. industrials, technology, mining etc). The figure contains a number of variables of the format "X *industry*" (e.g. DY industry, PTBV industry), these variables indicate normalization of the stock X's value in its industry sector – this is because it is less meaningful to compare these quantities between sectors due to reasons such as different risk levels inherent in business types. *Momentum* means the change over time of a quantity (e.g. DY industry momentum in the figure is a variable that measures the increase or decrease of DY industry over time). Further details on these topics may be found in the following sources [88, 70, 57, 41].

2.4 Adaptive Business Intelligence

In this section we discuss problem solving in the financial portfolio application using the paradigms of Adaptive Business Intelligence. Figure 2.3 shows a conceptual schema for system design of an the Adaptive Business Intelligence (ABI) system. Design elements are divided into three main components: optimization, prediction and adaptation.

Our approach combines a high fidelity problem representation and specific objective definitions with an adaptive methodology. The specification of the financial forecasting problem involves the construction of asset rankings, these are defined as ordered sets of assets based on valuation models that are optimized to take into account risk and return forecasts and present values. Rankings provide an evaluation of each assets worth, relative to the other potential choices within a defined portfolio setting. The relative ranking approach involves a comparative valuation problem and significantly contrasts with much previous work in intelligent portfolio management approaches. For example, the paper [88] describes time series prediction and so attempts to "time the market" or buy and sell assets by predicting future movement explicitly. The methodology of valuation leads to a comparatively holistic approach that enables a natural way for computational intelligence to be used in automating additional aspects of fund management as well.

The approach for trading is realistic and replicates and fits with the decision process of a standard fund manager (for instance, considering or allowing setting of frequency of portfolio re-balancing, transaction costs, liquidity constraints, etc). Although it is the case that the approach is also generic enough to be used for a wide range of financial assets, the focus is on equity classes as they provide an excellent

Name	Description
	Macro Data
INDEX	the market index, a weighted average by market cap of the value of listed equities.
RF RATE	interest rates for short term government bonds (3 months).
GOLD	the spot price of gold in USD.
OIL	the price of a barrel of crude oil in USD.
	Fundamental Data
DY	dividend yield for the company. A percentage value of the dividend income earned over the stock price.
PTBV	price to book value for a company. Literally calculated as stock value over accounting book value.
PE	price earnings ratio for a company. Calculated as the price of a stock divided by earnings per share.
PE2	a forecast of price earnings ratio for the next year by financial analysts.
MV	the market capitalization of a company. Calculated as the company stock price multiplied by the number of shares.
EPS	earnings per share.
TDE	total debt to equity ratio.
LDE	long term debt to equity ratio (> 1 year).
EBITDA	earnings before interest and tax.
ROA	return on assets.
ROE	return on equity.
	Technical Market Data
PRICE	daily close prices.
VOLUME	daily trading volume.

Table 2.1: Raw input types. The model factors are derived from these basic data types.

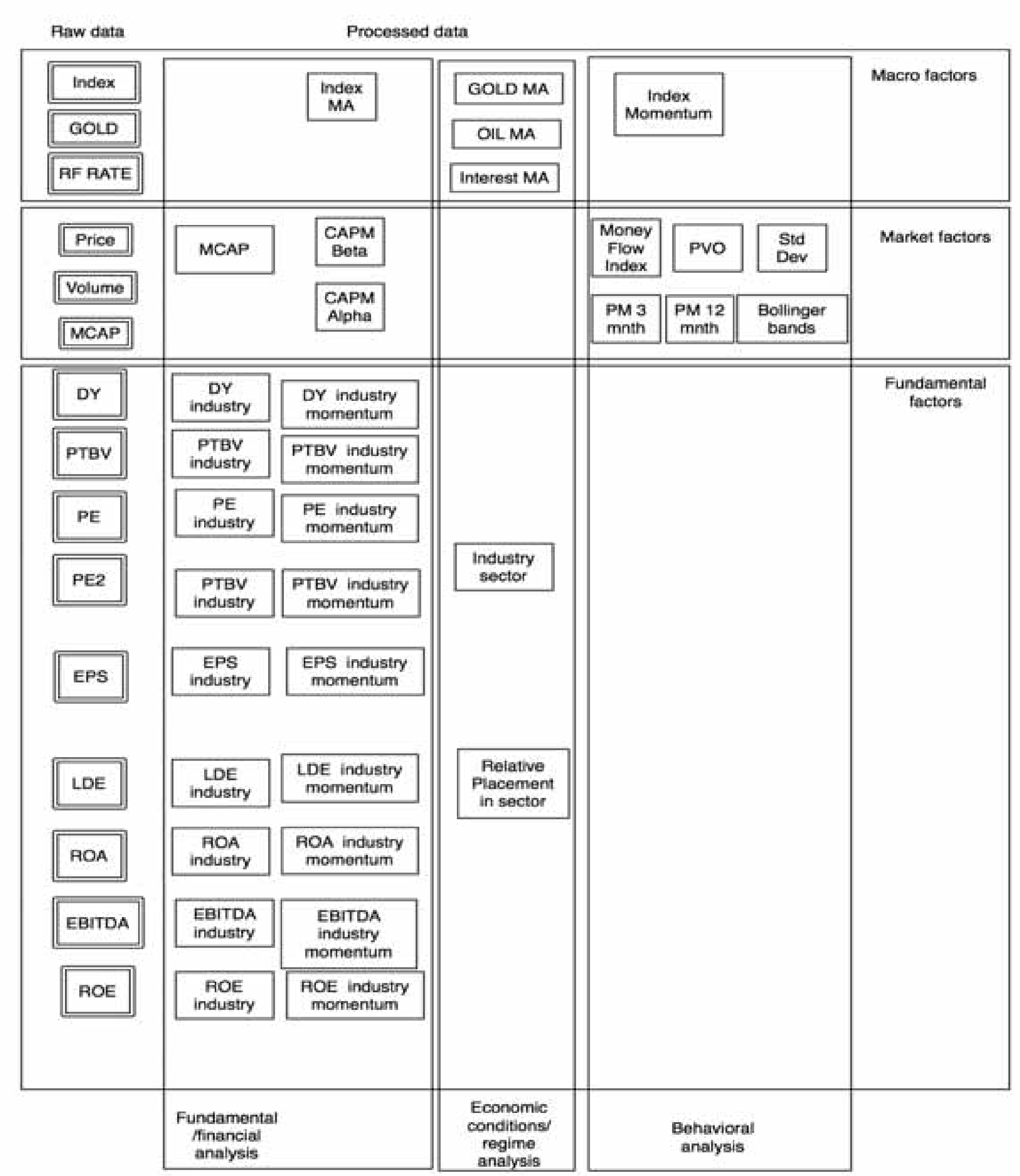


Figure 2.2: Multi factor stock market valuation model. See text for meanings of acronyms.

case for examining complex/dynamic/adaptive optimisation and have commercialization potential.

The problem specification is identified with respect specifically to finance. It is the case that the evaluation function/s are based on prediction of future values of some variables, so that the evaluation function *eval* is expressed as:

$$eval(\vec{x}) = f(\vec{x}, P(\vec{x}, \vec{y}, t)),$$

where $P(\vec{x}, \vec{y}, t)$ represents an outcome of some prediction for solution vector $\vec{x} = (x_1, \ldots, x_n)$ and additional (environmental) variables $\vec{y} = (y_1 \ldots, y_n)$ at time t. In multi-objective and constrained cases this expands to:

$$eval_1(\vec{x}) = f(\vec{x}, P(\vec{x}, \vec{y}, t)),\ eval_2(\vec{x}) = f(\vec{x}, P(\vec{x}, \vec{y}, t)), \ldots,\ eval_q(\vec{x}) = f(\vec{x}, P(\vec{x}, \vec{y}, t)), \tag{2.4}$$

subject to constraints: $c_i(\vec{x}, t) \leq 0, (i = 1, 2, , k)$. There are a number of approaches to incorporating these objective definitions computationally and in the context of an application. The incorporation may be either explicit or implicit. However it is always the case, in the problems we consider in financial markets, that these aspects of the problem are in existence in the real operating environment even if not explicitly in an abstract model.

In this thesis we consider the objective to be of this form and consider these issues using different objectives, penalties and other methods. We identify a number of important aspects to be considered for solving these problems. These issues include the following:

- a dynamic environment: time variable is present in the predictive model, $P(\vec{x}, \vec{y}, t)$, and the constraint model.
- a huge solution search space: The size of the solution space denoted by $\vec{x}$ above may be huge depending on the investment strategy developed.
- prediction based objective function: The usefulness of the objective function 2.4 is based to a large extent on the fidelity predictive model $P(\vec{x}, \vec{y}, t)$. This is distinct from existing research into classification and prediction where the predictive model is the final result of a search process and does not have the ability to quickly adjust to environmental changes.
- identification of environmental variables: The environmental variables are represented by $\vec{y}$ above and have a significant impact on the predictive modules

capability. Identification of methods to characterize the environment involves using domain knowledge (selection of economic indicators and so forth) and may also be the result of a separate search process.

The design and methodology we use is devised to consider these issues. In achieving steps towards this, contributions to extending existing research in finance and computational intelligence will take place. Existing approaches of quantitative methods in finance make use of computational power to quickly construct and test different (usually linear or regression) models. The new approach implements non-linear models that learn and adapt to the market environment over time. Further, some previous applications using CI to implement non-linear models, such as with artificial neural networks, have a focus on learning models fitted to a set of training data but with limited focus on using models over time and without an explicit consideration of adaptation and the dynamic environment.

The three key conceptual modules (prediction, optimization and adaptation) highlighted in Figure 2.3 have the following meanings:

1. Prediction: consideration of application forecasting aspects and development of the prediction module P with meaning as presented in Eq. 2.4,
2. Optimization: development, identification and tuning of search algorithms for model optimization,
3. Adaptation: developing methods for handling the time changing aspects (denoted by t in Eq. 2.4).

We focus on these main modules in the remainder of this chapter. Support modules including the Graphical User Interface, the Reporting and Database modules are not discussed in detail but we refer the reader to Chapter 9, which contains a description of a complete application for intelligent decision support. Out of sample testing and input data updates support a feedback loop that represents continual or periodic internal evaluation of the prediction models. Essentially, this distinguishes the adaptation mechanisms and allows the system to adjust to market conditions by responding to both recent input and evaluation of recent performance. Let us now discuss the three main modules in more detail.

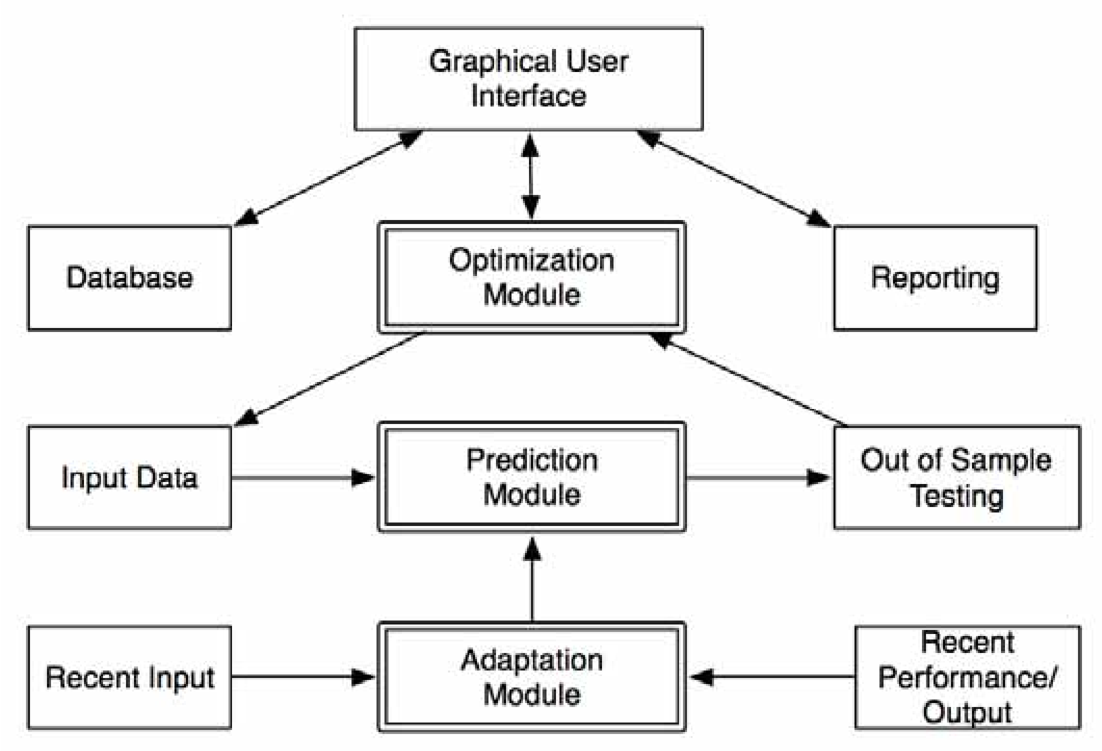


Figure 2.3: Adaptive Business Intelligence (ABI). The components for optimization, prediction and adaptation are the main logical divisions in design, each requires its sub components (and each other). The optimization loop controls the recommendation presented to the user, it is predominantly a function of the input data and the result of predictions that are updated when new data is loaded and in response to feedback from recent management portfolio performance.

2.5 Prediction

The prediction module, represented by $P(\vec{x}, \vec{y}, t)$ above, is one of the three key components of the methodology. The prediction task is quite complex as equity markets are difficult to value and forecast because they change with time and may be influenced by many variables whose impact is also time-varying. High fidelity solution models, $\vec{x}$, are potentially more capable of producing meaningful solutions and strategies that can perform in reality, given the complexity of the environment, by considering more information and varying the emphasis on different information types. The other component of the module, environment variables $\vec{y}$, can also be complex: they encode some information about the prediction environment (e.g. market regime, economic situation etc). The time variable, t, indicates the dynamic aspects of this model.

Solution representations that are interpretable and able to represent complex structures include fuzzy rules and grammatical structures. Such representations encode specific meaning from the application area. Examples of this research include [27, 88]; and an extended discussion of grammatical structures for trading may be found in [14]. The development of multi-rule based constructs provides a closer fit to the design of investment "styles". The development of more complex trading strategies may lead to better asset selection choices. Commonly, fund managers refer to "value" and "growth" strategies that involve selecting assets using several criteria. Representations through rules or grammatical structures can not only replicate this but also allow for the study of extensive deviations from the initial criteria that is considered.

2.6 Optimization

The optimization component of the approach involves promoting efficiency and effectiveness in seeking solutions within the search space, given the objective function and penalties etc. It is emphasized that, as indicated by the arrows in the diagram at the beginning of this section, the three modules are interrelated. Constraint handling is another aspect. These elements act to produce solutions with desired characteristics. There are several methods to implement hard (where solutions are required to satisfy them to be useful) or soft constraints (where satisfaction is balanced with other goals). These methods include decoders or repair operators, and penalties [28, 76].

2.7 Adaptation

The meaning and importance of the prediction and environment model parameters can change due to very fast shifts in the financial environment. In this application the ability to handle large input sets and optimize their utilization in a complex generated trading strategy is key, also to be considered is that the weighting and importance each input has on model development may change over time. Financial econometric techniques try to deal with this through regime-switching mechanisms [48, 112]. However, they are limited by parameter specification limitations and suffer from problems locating global optima due to the nature of the gradient-based search techniques that are employed.

Two specific classes of adaptation are distinguished. That of the system itself during optimization runs and application adaptation over time (such as from the financial environment and/or performance of the system relative to an external benchmark). The second type of adaptation includes various methods that control and change solutions used over time or to lead different to input data selection, focusing searches on specific areas of the search space containing solutions suited to present conditions and other methods.

Adaptation of system parameters and so on is an important issue. This applies in relation to parameters of the search algorithm and other problem specific parameters to influence training data, input variables, model specification variables and even environment variables. Choosing the right parameter values is a detailed task. There is a distinction between two major forms of setting parameter values: parameter tuning and parameter control. Parameter tuning amounts to finding good values for the parameters before the run. In parameter control settings are changed during the run. Attempts have been made to find the optimal and general set of parameters for tuning, however experimental evidence indicates that specific problem types require specific algorithm setups for satisfactory performance [77, 65]. Thus, the scope of optimal parameter settings is necessarily narrow. The approach used in this thesis incorporates methods for parameter setting during the run as well as predetermined tuning. By limiting direct setting of parameters sensitive to performance risks associated with data snooping [105, 101] are reduced. This is because there is a potential for setting parameters to attain high performance in historic testing that can fail in other periods.

Chapter 3

Literature Review

There are a number of papers that describe applications of CI (Computational Intelligence) related methods to financial tasks. In this chapter we survey this work. As research in this area has come from financial as well as computing science areas the discussion necessarily covers both sets of literature.

In Section 3.1 we list a range of applications of CI to portfolio and financial market related tasks. Section 3.2 provides discussion from the financial literature relevant to active investment styles and the efficient markets hypothesis; especially the relation of these to the approach to portfolio management used in this research. In some respects the goal of developing active trading strategies that consistently outperform the market is quite optimistic. This is mainly because of conclusions implied by the efficient markets hypothesis on the impossibility of out performing the market. However we establish a firm justification for the development of algorithmic trading strategies and also discuss some insights this type study can give into related financial research topics. To conclude, Section 3.3 provides past performance results achieved using CI and related methods for equity market trading from the literature.

3.1 Financial Application of Computational Intelligence

Computational intelligence techniques have been applied to analyze financial markets in different ways. Some approaches utilize unique characteristics inherent in these methods (such as the ability to optimize flexible representations) to approach problems in portfolio management in novel ways not feasible using widespread model optimization techniques used in finance, such as linear regression or even quadratic programming (see [43]). Other approaches involve application of methods such as genetic programming, as alternatives to traditional techniques use in finance: for example, to discover approximate solutions for difficult problems. Other applications involve integrating different components of important financial tasks or procedures using intelligent reasoning.

An influential paper, [45], provides a logical sequence of steps to consider for fuzzy linguistic decision analysis. They divide the task into the following: the choice of the linguistic terms (or inputs); the choice of the operator to aggregate these terms; and the subsequent choice of the best alternatives on the basis of the model. The final step, the application of the model, involves the *aggregation* of information found using the aggregation operator and the *exploitation* of the model by utilizing it to compare possible decision choices that involves ranking possible alternatives.

Evolving fuzzy systems make use of evolutionary computation to automate learning of a knowledge base that is represented using fuzzy logic [22]. The learning part may involve various components of a fuzzy rule or the rule base as a whole. A number of papers describe these methods in finance and prediction related applications. A method making use of evolutionary computation to tune fuzzy rules used in insurance underwriting is provided in [13]. Takagi-Sugeno-Kang (TSK) fuzzy systems [102] have often been used in prediction applications, for examples see [83, 46, 23, 110].

As an example of evolving fuzzy systems for prediction, a medical application that tunes many aspects of the knowledge base representation by applying an evolutionary algorithm in two stages is found in [23]. A predictive system that uses a TSK system to predict river dynamics is given in [46]. Very recently, a number of studies have appeared applying fuzzy systems specifically to predict financial data. For example [21] discusses application of a TSK style system to predict stock price series with the goal of minimizing forecasting error. They attained impressive accuracy results over 90%; the approach involved several stages to select model inputs and tune fuzzy rule parameters. These included regression analysis to select rule inputs and simulated annealing to tune rules. Also, [111] describes the use of a type-2 fuzzy logic system for modeling stock prices. Type-2 fuzzy systems, see [54], involve a fine grained representation as compared with type-1 fuzzy systems to enable modeling uncertainty in the linguistic descriptions. It is also possible to increase the complexity of the decision surface of a type-1 fuzzy system by adding additional membership functions, which may be of use in data intensive applications such as associated with the stock market.

A number of studies have been conducted using various knowledge base representations and learning algorithm for financial applications. We discuss some of these in the remainder of this section. As an example of using computational intelligence to develop novel knowledge bases, genetic algorithms are applied to extract knowledge from financial statements [104]. Examples of application of artificial neural network and fuzzy logic techniques for forecasting time series are found in [64]. A more complex system is presented in [60], which uses a hybrid of several CI techniques to predict financial time series, and promising results are obtained in tests using data from the Taiwan Stock Exchange. In a novel application of an evolutionary approach for learning models to anticipate the value of IPOs (initial public offerings) is described in [93]. Genetic programming is a very promising approach which has been used in a number of forecasting problems, in [86] and [87] the use of genetic programing to predict exchange rate volatility is examined. This is an important problem in finance with standard methods for approaching it. The natural computation methodology performs slightly worse than the comparative traditional approach by some measures but better in others.

Trading strategy design is an area where in recent years significant gains have been made in computational finance. A trading strategy is defined for this purpose as a set of rules for making trading decisions including recommendations to buy or sell etc and also aspects such as exit strategies or position holding periods. Some research in this was published in the 1990s (especially by financial academics) when techniques such as neural networks and genetic algorithms became well known. In general, the applications used in these tests were relatively simple in comparison with more modern systems because of the area of expertise of these researchers outside computer science and also the early stage in the development of the strategy generation techniques. This work generally involved an optimization of a single rule or parameters for a rule, for examples see [84, 9, 2]. These simpler approaches met with mixed success.

More recently, research applying techniques that use quite complex systems: for example to learn multiple rules, complex non linear relationships and exotic model inputs data and problem specifications. In [3] a particular adaptive neuro-genetic algorithm is described that provides substantial returns for intra day trading. Many of these more complex approaches and applications to problems not considered for computational modeling show quite good results. For example [62, 103] discuss the use of accounting information and natural language data from news feeds and internet message traffic is used profitably in stock price prediction. Trade execution is another area where CI has been applied. This activity involves processes such as splitting a large trade into smaller trades, timing of sending an order to a stock broker, the type of order etc with the goal of optimizing market impact and minimizing the overall resulting cost of a transaction, see [66]. This is a very interesting area which does not necessarily involve forecasting where traditional optimization could yield profits or cost savings for traders.

Portfolio optimization involves selecting optimal stocks and weightings in a portfolio under constraints for diversity and others. This is an area that has been a focus of considerable research. A classical (financial) technique to specify the problem is the Markowitz mean-variance framework. Research such as [41] use this model optimize risk and return relationships. Portfolio optimization is a high dimensional, constrained optimization problem. In [5] the problem is represented using a "scenario tree" of future possible portfolios to anticipate future portfolio developments after particular adjustments. They use a multi objective algorithm with a number of constraints. Further examples of multi objective optimization for this task are provided in the following papers [19, 92, 94]. Differential evolution is applied to manage a constrained index tracking portfolio (a portfolio designed to replicate an index value usually containing fewer stocks than the whole index) is discussed in [72], the optimization problem is to set weightings for the portfolio elements.

3.2 Active Trading Strategies

Developing trading strategies that signal when investors should buy or sell certain financial instruments and how it is possible to do so is a significant research topic. Research in this area has received greater attention as an appreciation for the ease by which computational algorithms can develop complex trading strategies is further realized. Research such as described in [68] and [2] highlight the possibilities for evolutionary computation to provide trading strategies, based on pattern recognition, to profit from equity market trading. Papers in academic finance journals primarily focus on examining how well genetic algorithms can develop specific trading rules using historical prices to test, ex post, their profitability.

This type of research is directly related to the study of market efficiency. In an efficient capital market it would not be possible for traders to make a profit from past data as all relevant information for pricing a security today would be incorporated in today's price (there are different levels of efficiency, "strong form" efficiency asserts even insider trading can not attain outperformance on average).

Therefore, many finance papers (see [49], [84], and [34] for example) inter-relate the issue of market efficiency with the ability for genetic algorithms to literally "beat the market". Results are somewhat mixed. Although there is general consensus that financial markets do sometimes exhibit periods where certain trading rules work (see [16]), it is hard to find clear evidence that a single trading rule can function over an extended period of time. This could be due to the fact that financial markets are ever-evolving, and in fact given the number of analysts that are employed in all the major financial trading institutions, when a trading rule is found to work it would not take long before it is exploited until it no longer yields a significant profit. It is therefore more promising to take an adaptive approach to see if trading rules can be constructed that also continually evolve as the markets change. In addition, it has been found that simple strategies an be enhanced by increasing their complexity [49].

A number of studies, see [71, 16, 12, 20, 84, 49, 18, 17, 35, 69], contain evidence that rule based strategies can provide information to help determine future price movements. A significant amount of this literature examines momentum strategies — stocks that performed well recently often continue to in the future. Price momentum is considered to be a "fourth" factor in the Fama and French three-factor pricing model, see [32]. Several reasons have been put forward to explain why inefficiencies can occur as implied by empirical observations of outperforming active strategies. Predominantly these involve attributing "irrational" human behavioral factors that can make the market inefficient (see [7]). Technical trading rules may also be coincidentally picking up institutional trading traits, such as market depth (see [58]).

There is, however, also a corresponding list of research that has shown technical trading is not a profitable strategy, particularly after transaction costs are considered. This includes early work by Fama and Blume [31], as well as [2, 95, 85, 11, 75]. One argument that warrants special attention is that any tests conducted on the effectiveness of technical trading rules run the risk of the results being prone to data-snooping. Simply put, technical trading rules that are popular today are so because they worked in the past. Another important data collection issue exists regarding "survivorship bias" which can also make matters worse if indices are used without accounting for the impact when failing stocks de-list. These issues are treated in [105] where these pitfalls are described in detail. Recent research into the success of trading strategies now specifically exerts greater effort to ensure results are not effected to these problems; as a result any research from a computer science perspective should provide adequate performance analysis to be take seriously. Some recent research that focuses on this is provided by [50, 90, 67].

In the above literature, focus is placed on the success of utilizing one specific type of rule in isolation to others. A moving average rule, for example, might be examined with varying moving average lengths and tested on a series of indices or stocks. However, a cursory examination of any of the currently popular technical trading books or websites that investors read regularly cite that one should never place complete faith in a single trading rule, and instead check for confirming signals from other indicators (for example see [1]). A technical trader, therefore, is unlikely to trade using just one rule and would instead utilize a combination of chosen rules to determine when to buy or sell a stock. The problem from an academic point of view is that it becomes considerably difficult to test the success of such a strategy where there are an endless supply of combinations of rules that could be put together to determine a functional trading strategy. Nevertheless, by-and-large, technical traders do exactly this, choosing a certain set of rules to determine trades. According to the *Merrill Lynch Institutional Factor Survey*, mommentum based (technical) indicators were among the most popular valuation indicators used by investement managers in practice during the period 1989 to 2001 [100]. In the research approach of this thesis, heuristic search methods combined with rule base encoding methods provide the capability to quantitatively implement a more realistic (as compared with real application by traders) method for finding and adjusting trading rules.

3.3 Related Results

There have been several studies showing that evolutionary computation and other algorithmic trading methods may be successfully applied to discover trading rules that yield positive results using heuristic searches and various nature inspired algorithms, for example [52, 88, 15, 37]. The benefits of adaptation where trading rules are updated as the market changes, see [15, 37], is also supported by research such as [16]. It appears to be the case that it is a much easier task to find particular rules that work over limited periods rather than rules that perform at all times. For additional examples see [53, 88, 68, 89]. A key benefit of using adaptive computationsal intelligence systems for market trading is the provision of means to discover and exploit rules when they work and then as time progresses discard them to be replaced by new working rules. In this way a forecasting model is defined not so much as a particular set of rules or formulas but rather by parameters and inputs for a hueristic search.

A technical indicator index trading system that uses grammatical evolution is presented in [88]. The method was used succesfully in comparison to buy and hold and in some cases the marked index for a test period from 1984 to 1997. The grammer included a number of technical indicators derived from price and volume data. Trading signals to either Buy, Sell or Do Nothing were derived by post processing an evaluation of the formulas derrived by GE. The fitness evaluation used in the evolution process was as follows: $fitness = (return - x(maximum\ cummulative\ loss))$. The fitness objective considered both risk and return. The solutions were used to trade a particular market index using previous data for the index series and was tested on FTSE, DAX and NIKKEI data sets. The best out of sample returns were higher than a benchmark of holding the index on the FTSE and NIKKEI data sets.

In a following paper, see [15], the authors of [88] discussed above provided several extensions to their approach including expanding the set of technical indicators used in the grammer and using methods to adapting strategies to recent data using a moving training data window. It was reported that the adaptive approach was able to beat the Nikkei 225 Index by 20.85% in one of the tests over a period starting in October 1992 and ending in December 1997.

An application of a population based algorithm to find rules for long and short stock selection is found in [109]. This system used trading simulation to evaluate solutions (non-linear equations). A single objective combining risk and return based on the Sharpe ratio was used in solution evaluation. They used a set of inputs that included technical indicators, raw price series, some fundamental data and exchange rates. An emphasis was placed on finding robust solutions, results for out of sample performance in the emerging Malaysian stock market were better than an index and also non-evolving technical analysis strategies.

Chapter 4

Initial Implementation

This chapter describes the initial implementation of an adaptive asset allocation strategy. The system adjusts to dynamic market conditions. The methodology will be extended in later chapters. This approach was published here [37]. Section 4.1 describes the representation of the forecasting models involved. Section 4.2 the process to search and optimize strategies. Section 4.3 describes the evaluation procedure (for comparing the performance of candidate strategies) which makes use of a trading simulation in historic data. Finally, Section 4.4 introduces mechanisms for adapting the prediction model in the dynamic operating environment.

4.1 Representation

The fuzzy rule base representation, in combination with the evaluation method discussed in Section 4.3, facilitates simulation of (human) financial reasoning on the basis of financial research (as understood by financial analysts). By evolving natural language rules, the problem representation promotes a process analogous to searching possible "human expressible" decision models. After assigning semantic meanings to data, the search algorithm tests formulations of possible rules, corresponding to a vast number of possible *meaningful* semantic expressions, according to how well they performed the task in historical data.

Through this approach we are able to avoid many potential problems and additional complexity that is associated with quantifying, explicitly, numerous cases of qualitative variation between different categories of data and also changing data distributions as new information (data points) is read. Advantages, of the linguistic representation, are obtained both in the specification and design of the system; and also in the computational search for prediction models, as the scope of this search is largely controlled by selection of linguistic variables (see below).

The fuzzy rule base representation we use enables intuitive natural language interpretation of trading signals and implies a search space of possible rules that corresponds to trading rules a human trader could construct. An example of a typical technical trading rule such as "buy when the price of a stock X's price becomes higher than the single moving average of the stock X's price for the last, say, 20 days" (indicating a possible upward trend) could be encoded using a fuzzy logic rule such as "If *Single Moving Average Buy Signal* is *High* then *rating* is 1"; conversely we could have a trading rule such as "sell stocks with high volatility when the portfolio value is relatively low" encoded by a fuzzy rule: "If *Price Change* is *High* and *Portfolio Value* is *Extremely Low* then *rating* is 0.1".

Each fuzzy rule base consists of a set of *If-Then* rules where the "If" part specifies properties of technical indicators and the "Then" part specifies a rating with 10 discrete levels given a stock with these properties. The rule inputs are termed linguistic variables in the fuzzy logic component. Clearly, at least one linguistic variable must be defined to construct rules. We use $V = 9$ linguistic variables, Section 6.4 describes each of these linguistic variables used are described. The output is interpreted as a rating of the strength of a buy recommendation given fulfillment of the If part. It is possible for the If part of a rule to refer to any combination of the technical indicators the system uses to give one output rating. A rule base may contain at least one and no more than $O = 30$ rules.

The value of each linguistic variable is described by one of a possible seven fuzzy membership sets. These are defined describing the relative magnitude of a particular observation: *Extremely Low* (*EL*), *Very Low* (*VL*), *Low* (*L*), *Medium* (*M*), *High* (*H*), *Very High* (*VH*), and *Extremely High* (*EH*). Membership functions map crisp data observations to degrees of membership of these fuzzy sets.

The membership functions are initialized using historical data. First the fuzzy membership sets for each variable are constructed by sorting a series of observed data values from low to high. The range of values that have a degree of membership greater than 0 is set to cover the same number of observations. If data for constructing the membership functions contained 70 observations for a linguistic variable then each membership function for that variable would contain 10 observations: each linguistic variable is associated with seven fuzzy membership sets (*EL*, ..., *EH*). The centre of each membership function is the mean of the observations that lie between its *min* and *max*. Whenever the system observes new data the

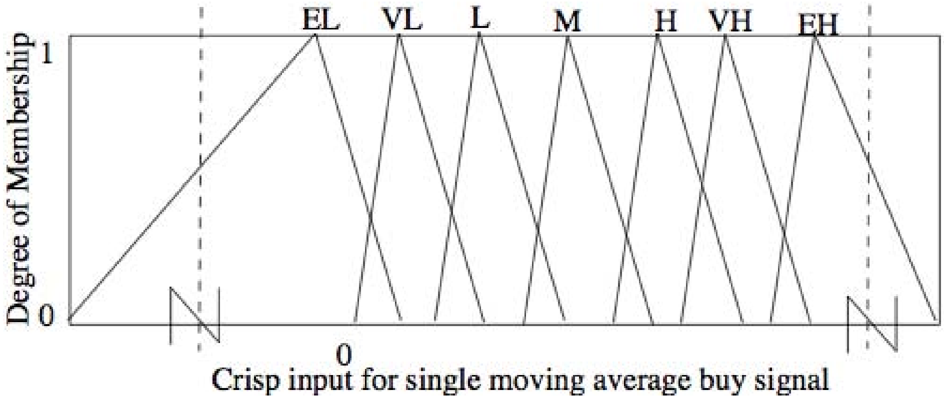


Figure 4.1: Sample membership functions for fuzzy sets *EL*, ..., *EH* extracted from the single moving average buy signal linguistic variable.

membership functions *min*, *max* and *centre* are updated.

Figure 4.1 shows a visualization of the membership functions for the single moving average indicator. The membership functions are triangular: the mapping from an observation to a degree of membership for each membership function (*EL*, ..., *EH*) is fully defined by specifying the sets *min*, *centre* and *max*. The lowest, *min*, and highest, *max*, specifications refer to the lowest and highest variable observations that are the extreme values that are members in the set with the least degree; the *centre* belongs to the membership set with the highest degree (the top of the triangle in the visualization in Figure 4.1). The meaning of these definitions are also illustrated in Figure 4.2, in this image the meaning of the membership sets in relation to input data is shown clearly. The vertical axis shows the size of a price oscillator indicator calculated from daily prices. This indicator is generally interpreted by traders to emphasize cyclical trends in stock prices: it implies a signal to buy a stock when it trends to lower values because this means the stock over sold and a signal to sell when the indicator is in the higher range. Depending on particular cases the meaning of higher or lower range can vary. The right hand of the figure shows fuzzy membership functions used to define these values by the system. A large number of time series for different stocks are compared efficiently in this way.

Any "If" part may include up to $V = 9$ linguistic variables; each linguistic variable can take one of seven possible values; the output for each rule gives one of 10 different ratings; there can be up to $O = 30$ rules in each rule base. This implies a total number of unique rule bases (phenotypes) in the order of 10^{270}.

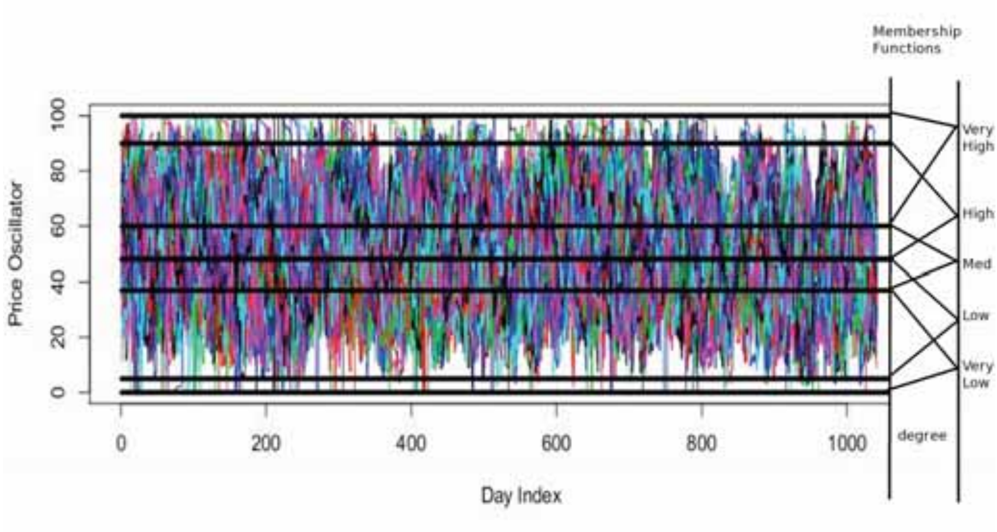


Figure 4.2: Classification of over 200 input data series (one for each stock) over five years using fuzzy descriptions.

An example of a phenotype rule base that could be produced by the system is given below. It consists of three rules; each rule has one or three linguistic variables in the "If" parts:

- If *Single Moving Average Buy Signal* is *Extremely Low* then *rating* = 0.9.
- If *Price Change* is *High* and *Double Moving Average Sell* is *Very High* then *rating* = 0.4.

1	2		3		4		5		6		7		8		9		10		11
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F

Figure 4.3: Internal rule base representation for a rule base with $O = 5$ and $V = 9$. B indicates a boolean value: $B \in \{T, F\}$; I an integer: $I \in \{1, 2, \ldots, 7\}$; and F a float: $F \in \{0.1, 0.2, \ldots, 1.0\}$.

Internally, each rule is represented using a sequence of slots. With reference to Figure 4.3: Column 1 contains a Boolean value to indicate whether the rule is active; Columns 2 through to 10 represent the rule inputs (each corresponds to a linguistic variable) and contain (a) a Boolean value indicating whether or not the linguistic variable is active, and (b) a number from 1 to 7 representing a membership function for the variable (the integer 1 corresponds to *Extremely Low* and 7 to *Extremely High*); finally, Column 11 indicates the rule output rating and contains a single floating point value from the set $\{0.1, 0.2, \ldots, 1.0\}$). The internal representation for a rule base is simply a 30×11 matrix (note that Columns 2 – 10 contain two values, a Boolean and an integer). It follows that for $O = 30$ the number of possible internal rule base representations (genotypes) is of the order 10^{338}.

As an example, the genotype representation of the phenotype given above is provided in Figure 4.4. Note that for compactness the illustration is of a rule base with $O = 5$ rules, for $O = 30$ the rule base additional rows would have false values in the first slot.

- If *Single Moving Average Buy Signal* is *Extremely Low* then *rating* = 0.9.
- If *Price Change* is *High* and *Double Moving Average Sell* is *Very High* then *rating* = 0.4.
- If *On Balance Volume Indicator* is *Extremely High* and *Single Moving Average Buy Signal* is *Medium* and *Portfolio Value* is *Medium* then *rating* = 0.5.

T	F 2	T 1	F 7	F 1	F 6	F 1	F 7	F 4	F 1	0.9
F	F 4	T 2	F 2	T 3	T 4	F 5	F 1	T 6	F 2	0.3
F	T 1	F 3	F 2	F 2	F 2	T 4	F 4	F 4	F 4	0.7
T	T 4	F 5	F 1	F 4	F 2	T 5	F 2	F 2	F 5	0.4
T	F 6	T 3	F 3	T 3	F 7	F 3	T 7	F 1	F 3	0.5

Figure 4.4: Example of the internal rule base representation. The order of the columns indicates the particular linguistic variables, both this order and the meaning of the variables is given Section 6.4.

4.2 Evolutionary Process

The fuzzy rule bases undergo an evolutionary process. An initial population of rule bases (genotypes) is selected at random and may be seeded with some rule bases that correspond to accepted technical trading strategies, for example the seeds used in the experiments discussed in this paper are given in Section 6.4 of this paper.

The evolutionary algorithm used in our asset allocation system is summarized by the following sequence of steps:

1. Initialize population P of n solutions (each solution RB_i is a rule base):

$$P = \langle RB_1, RB_2, \ldots, RB_n \rangle ,$$

2. Evaluate each solution: calculate $eval(RB_i)$ for $i = 1, \ldots, n$,

3. Identify the best solution found so far ($best$),

4. Alter the population by applying a few variation operators (tournament selection of size 2 is used),

5. Apply a repair operator to each offspring; this operator controls diversity of offspring with respect to the best solution $best_{previous}$ from the previous generation (elitism is not used),

6. Repeat steps 2-5 successively for N generations,

7. The best solution after N generations represents the final solution.

Three variation operators (one mutation and two crossovers) and one repair operator are used in the process.

The mutation operator works by possibly modifying each gene of a single parent rule base in the process of producing an offspring. The type of gene remains the same: for instance, a Boolean value cannot become an integer used to represent a membership function nor a decimal used to represent an output rating. If a gene is Boolean it is flipped. Otherwise, if it is an integer or float, one of three events occur with equal probabilities:

1. The corresponding gene in the parent is incremented or decremented (equal probability for either)by a small amount, δ, to derrive the offspring gene: for floats $\delta = 0.1$ and for integers $\delta = 1$. Since integers represent membership sets the change corresponds to a shift of one degree of membership (for example, from *Low* to *Very Low*).

2. The gene in the offspring is assigned a new value at random. For an integer gene the new value is selected from the domain $1, 2, \ldots, 7$ and for a float from the domain $0.1, 0.2, \ldots, 1.0$.

3. The corresponding gene in the parent is passed unaltered to the offspring.

The two crossover operators combine genes from two parents to produce a single offspring. The first one, uniform crossover, assigns each gene in the offspring the value of a gene selected from one of the parents (the parent that provides the gene value is selected with equal probability). The second crossover operator assigns the rows of the offspring matrix by selecting — with equal probability — rows from both parents. In other words, the effect of this operator is to build a new rule base by choosing complete rules from each parent. Finally, a repair operator is used to maintain rule base stability between generations. It takes two rule bases and "repairs" the genotype of the first to be no more than p percent different from the second genotype.

4.3 Evaluation of a Fuzzy Rule Base

The evaluation process comprises of three stages: in the first stage individual stocks are evaluated according to a rule base (Section 4.3.1); in the second stage, the overall rule base's performance is evaluated (Section 4.3.1). The return on investment (ROI) is adjusted in the final stage of the evaluation process (Section 4.3.2).

4.3.1 Rating of individual stocks

In this section the procedure to assign a rating to stocks with respect to a rule base is explained. For any stock X a rating $RB(X)$ is defined. This mapping will be described using an example. Consider a rule base as follows:

1. If *Single Moving Average Buy Signal* is *High* then *rating* $= 0.7$.

2. If *Price Change is High* and *Volume Change* is *Very High* then *rating* $= 0.4$.

On a particular day t the following observations are made of technical indicators for stock X:

1. *Volume Change* = 0.5
2. *Single Moving Average Buy Signal* = 0.95
3. *Price Change* = 0.2

The first step of the process is to process each rule individually. First consider the single If component of the first rule:

- If *Single Moving Average Buy Signal* is *High*

We observed that for stock X on day t the value for *Single Moving Average Buy Signal* was 0.95 on day t. We must find the degree that this observation is *High* to see how much it matches the rule: the membership function for *High* is defined by its min, centre and max which are, in this case, 0.12, 0.97, and 3.88 respectively. Using Equation 4.1, a membership function defined by these values maps the observed value 0.95 to a degree of membership of 0.97 in *High* or 97% *High*, a visualization of this procedure is given in Figure 4.5.

$$m(x) = \begin{cases} \frac{x - min}{center - min} & , \text{ if } min \leq x \leq center \\ 1 & , \text{ if } x = center \\ \frac{x - max}{center - max} & , \text{ if } center \leq x \leq max \\ 0 & , \text{ otherwise} \end{cases} \tag{4.1}$$

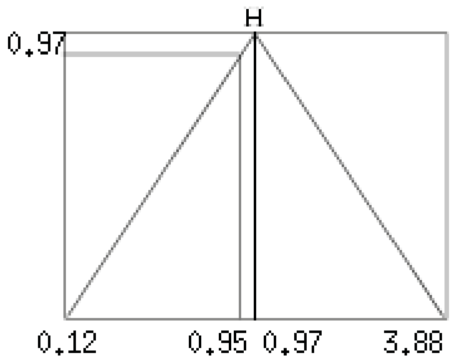


Figure 4.5: Finding the degree of membership of observed *Single Moving Average Buy Signal* 0.95 for stock X is *High* with degree 0.97.

Since the first rule only has one “If” part we now consider the output rating part of the rule: then *rating* = 0.7. Recall from Section 4.1 that *The output rating is interpreted as a rating of the strength of a buy recommendation given the total fulfilment of the If part*. By applying the membership function the degree that a rule fulfils the “If” part is found: the rating is adjusted proportionally to the degree of membership of an observation to the linguistic variable specification in the “If” part. As the rule fulfilled the “If” part of the rule to the degree of 0.97 we adjust the output rating: $0.97 \times 0.7 = 0.679$.

The system looks at each rule in turn, the second rule in our example is slightly different because it has two inputs which must be combined using a fuzzy conjunction operator:

- If $\widehat{\text{P}}$rice Change is *High* and *Volume Change* is *Very High* then *rating* = 0.4.

Initially, each term in the rule is processed separately. Using the process elucidated above for the first rule; it is determined that the observation *Price Change* = 0.2 implies membership in the fuzzy set *High Price Change* = 0.5; and that *Volume Change* = 0.5 implies membership in *Very High Volume Change* = 1. These two values are combined using a common fuzzy *and* operator: multiplying the membership degrees. Hence the combined membership: $0.5 \times 1 = 0.5$. In the same way as for the first rule we adjust the output rating: $0.5 \times 0.4 = 0.2$.

The final step of the process is to derive an output rating for the whole rule base, $RB(X)$; this rating combines the results for each rule to give a rating for stock X given some input data. Recall that for the first rule the result was 0.679 and for the second it was 0.200. To get the output rating the center of mass of the results from each individual rule is found. In the example this value is $(0.679+0.2) \div (0.7+0.4) = 0.799$.

$$RB(X) = \frac{\sum o_i}{\sum r_i}, \tag{4.2}$$

where o_i is the output of rule i for stock X, and r_i is the rating of rule i.

Evaluation of rule base performance

Using the procedure explained in the previous section for stock X a rule base is applied to each stock in the market. The result is a ranking of all stocks in a market M that is ordered by rating:

$$R(M) = \langle X_{i1}, X_{i2}, \ldots, X_{in} \rangle, \tag{4.3}$$

where $M = \{X_1, X_2, \ldots, X_n\}$ and $RB(X_{ik}) \geq RB(X_{ik+1})$.

The performance of a rule base RB is measured through analysis of the results of applying RB to simulated trading. The ranking of stocks discussed in the previous section that is implied by each rule base contains the information used in trading. A decoder defines the interpretation of the ranking to make decisions for portfolio construction.

The simulation takes place over a set period of time — a window of historical data. In the simulated scenario an initial capital is allocated to which is the used to construct an initial portfolio on day 1 of the simulation period. This initial portfolio is updated and traded over the rest of the data window. The decoder (see Figure 4.6) formulates buy and sell decisions given a ranking for trading the portfolio.

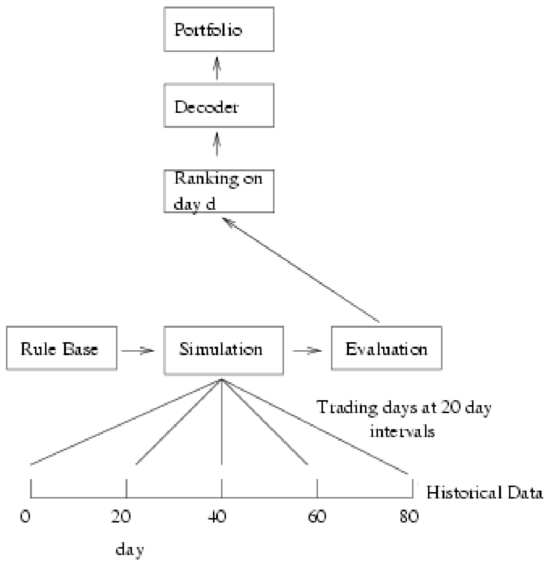


Figure 4.6: The decoder takes a ranking and recommends a portfolio.

In the system, a portfolio P_t is defined as a vector of holdings of stocks in $M = (X_1, \ldots, X_n)$ at time t:

$$P = [a_1 X_{i1}, a_2 X_{i1}, \ldots, a_k X_{i1}], \tag{4.4}$$

where $a_1, \ldots, a_k$ are natural numbers, $\{X_{i1}, \ldots, X_{in}\} \subseteq M$, and $Value(P, t) = \sum a_m \times price(X_{im})$.

Two key parameters used in the decoder. They are *buy best stocks percentage* and *Sell Worst Stocks Percentage*. *Buy Best Stocks Percentage* is the percentage of stocks to select from top of the ranking and *Sell Worst Stocks Percentage* is the percentage to sell from the bottom of the ranking. In all simulations we used a value of 10 for both parameters.

The process for updating a portfolio P_1 to get the next portfolio P_2 involves creating a new ranking on trading day 2 and selling stocks held that are at the bottom of the new ranking up to *Sell Worst Stocks Percentage*. Using the cash from selling the worst stocks as well as any unallocated cash the top *Buy Best Stocks Percentage* of stocks in the new ranking are bought if they are not already in the portfolio. Cash is distributed evenly over the best-ranked stocks until cash either runs out or the portfolio contains the *Buy Best Stocks Percentage* of all stocks in M. Trading does not usually take place every day in the simulation scenario, portfolio updates are processed at set intervals defined by the distance between trading days (distance between 20 day intervals shown in Figure 4.6).

Transaction costs are accounted for in the simulation. For each transaction the transaction cost is deducted from the capital. The portfolio is updated after a set number of days d (typically every 20 days), and then every d days after that.

Performance is highly influenced by the assumptions made in the simulation including the method used to interpret the ranking by the decoder and the parameters used to guide the portfolio construction during the simulation where the parameters are tied to each rule base and may be subject to the evolution with the rules. Another crucial parameter is choice of the historical data window and we discuss this aspect below in Section 4.4.1.

Rule base performance is evaluated by analysis of portfolio performance during simulation . The measure used for evaluation of portfolio performance is Return on Investment (ROI) during the whole simulation period (see Equation 4.5).

$$ROI = \frac{e^{\ln(V_{t1}) - \ln(V_{t_0})}}{t_1 - t_0}, \tag{4.5}$$

where V =Portfolio Value, t_1 = End Time and t_0 = Start Time.

The result of the simulation is the ROI during simulation for $RB(X) : ROI(RB(X))$. To compare $RB(X)$ to another rule base $RB(Y)$ it is the case that if $ROI(RB(X)) > ROI(RB(Y))$ then $RB(X)$ is better than $RB(Y)$. This basic criteria is supplemented by a few additional characteristics of performance that are considered in the final evaluation described in the next section.

4.3.2 Final evaluation

Additional criteria are considered by the evolutionary algorithm as well as the ROI fitness measure when measuring performance. This is implemented using penalties that reduce the fitness of solutions with certain properties. These are used to guide the evolutionary search away from rule bases that produce undesirable return distributions within the training period (even if the return over the whole period is good) and also to prevent over fitting solutions to training data. The final evaluation value equals the ROI in simulation minus penalties. There are two penalties applied to modify ROI, and they are:

1. Portfolio loss penalty
2. Ockham's razor penalty

Let us discuss each penalty in turn, starting with the portfolio loss penalty.

In the simulation evaluation we measure portfolio return on each trading day, as well as the final return on investment over the simulation period. Solutions that result in a reduction of portfolio value (during simulation) are penalized if they result in losses on any trading day even if at the end of the simulation period the return was high (see Equation 4.6). This mechanism provides a risk reduction facility and by adjustment of the penalty values that are imposed lever to focus the search for rule bases that can give particular return characteristics. The penalty becomes progressively higher for large losses.

$$m(x) = \begin{cases} 0.01, & \text{if} \quad \delta \leq -5 \\ 0.1, & \text{if} \quad -5 \leq \delta \leq -10 \\ 10, & \text{if} \quad \delta \leq -10 \end{cases} \tag{4.6}$$

where δ is the change in portfolio value since the previous trading day.

For example, if we had a 120-day simulation with a trading interval of 60 days the penalty would be applied twice: once at 60 days and once at 120 days. In this example if a rule base had an initial value of $\$10,000,000$ on day 1 of simulation then at day 60 a value of $\$95,000,000$ and on day 120 a value of $\$99,500,000$ the penalty would be calculated on each trading day as follows:

1. On day 1 no penalty is applied as it is the first day,

2. On day 60 the penalty is incremented by 0.1 because the portfolio lost 5% of its value (Equation 4.6),

3. On day 120 no penalty is applied because the portfolio increased 4.5% since the previous trading day.

The second penalty, Ockham's razor (Equation 4.7), reduces the fitness of solutions with many rules unjustified by returns. The reason that it is better to have fewer rules is that this encourages generality rather than over fitting to training data.

$$P_{ockham} = \text{ number of rules } \times k, \tag{4.7}$$

where k is a penalty constant.

The penalties are added together to get an overall value for each single rule base. This value is deducted from the ROI for that rule base. Figure 4.7 gives an overview of the process required to determine a fitness value comprising a penalized ROI value for each rule base.

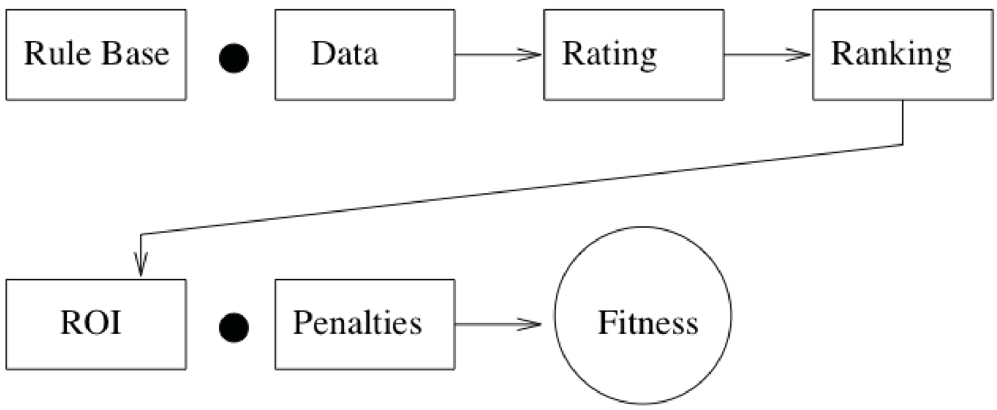


Figure 4.7: The sequence of steps and operations involved in the evaluation process.

Using the methods to set the objective of the EA the characteristics of rule bases with higher performance from both a risk and return perspective are targeted. The result is a best rule base that is able to be used for real trading. It is important that the rule base is applied to real trading in the same way as in simulation.

4.4 Adjusting Solution

We influence the search process by selecting of training data periods to consist (solely) of recent periods and by maintaining a memory between different optimization runs.

4.4.1 Data window

During the search process the performance of rule bases is evaluated based on data as described in Section 4.3. Rule bases that perform well during the training data window are identified by the search. We first discuss the methods to select data windows and then controlling the search.

Three methods for selecting a data window are considered:

1. Initial Window,
2. Extending Window,
3. Sliding Window.

The initial window (Figure 4.8) uses a single initial period to evolve a rule base and then the rules from this period are used for all future trading. The extending window (Figure 4.9) uses all the historical data available to evaluate rule bases. The sliding window (Figure 4.10) uses a recent historical time window for evaluation. In methods 2 and 3 the rule base adapts to consider the changing market, the sliding window fits the rule base to a period in the recent past. Note that in 2 and 3 the rules are applied to trading immediately after the last historical data period has transpired. Another approach to be tested in the future will involve identifying characteristics of the market (market regimes) during training windows and then applying rule bases when the market appears to be exhibiting these characteristics.

4.4.2 Memory of previous solutions

The system makes use of a repair operator to keep a memory of the best solutions from previous runs and focus the search close to the best individual from the previous run. A new search takes place for each new window in the extending and sliding window methodologies. However, instead of starting with a completely new population a memory is maintained of the best solution from previous windows. This is

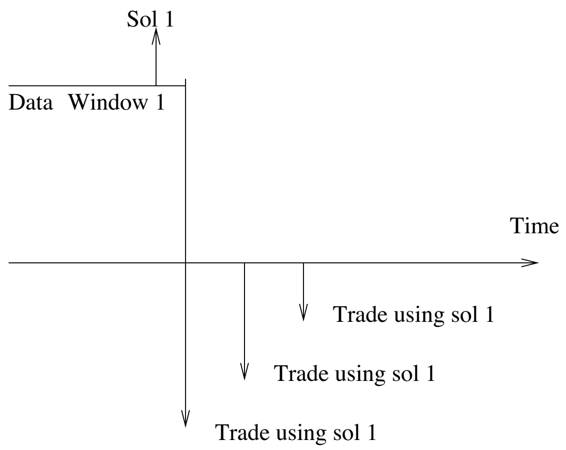


Figure 4.8: A static rule base approach.

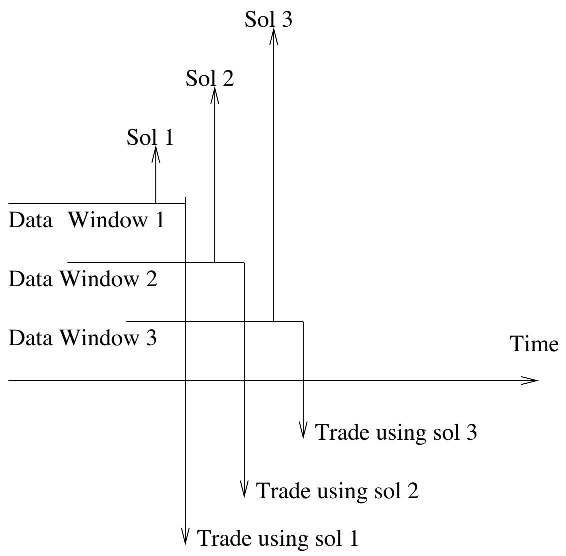


Figure 4.9: A sliding window approach to adaptation by updating training data.

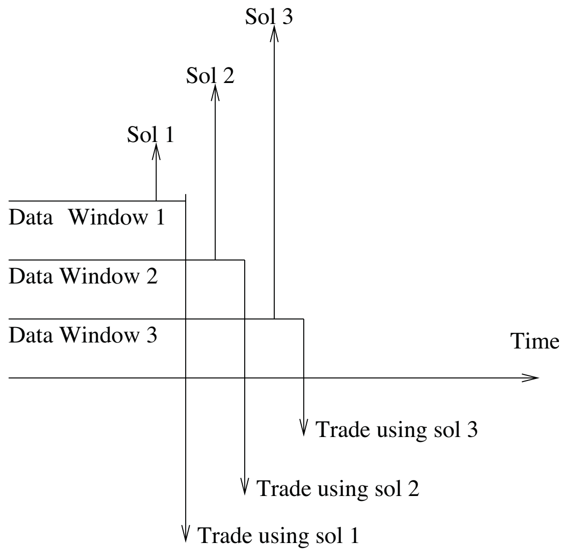


Figure 4.10: An extending window approach to adaptation by updating training data.

achieved using the repair operator that enforces a condition on new individuals generated that they have a percentage of identical genes to another specified individual (Section 4.2). The best solution from the previous window is used in the generation of the initial population for each window and the percentage of same genes for the operator is set by a fixed parameter at the start of each run.

There are several important reasons that this mechanism is used. The most important is that it is desirable that subsequent solutions should be a similar as possible to maintain a stable investment strategy over time except when the underlying data processes change. This serves to minimize transaction cost. In addition, the run time of each run is shortened by focusing the search close to previous best and unless a new optima is found the generations without improvement stopping condition is reached.

4.4.3 Rationale for the design

It is intuitively a very probable hypothesis that the processes influencing market prices change over time. This has also been investigated in behavioural finance, see literature review. So an adapting approach to modelling is likely to yield benefits.

Compared with regression, advantages provided by using CI techniques in valuation include:

- modelling the time changing impact of different model input variables or factors;
- using a non-linear solution representation;
- and, interpreting huge amounts of data constantly updated online.

Compared with other soft computing techniques such as Artificial Neural Networks and Grammatical Evolution, fuzzy systems are interpretable by humans in the sense of being natural language statements with a logical semantic structure.

Grammatical evolution and neural networks have been used in computational intelligence applications in this problem domain (for example see [88, 15, 84, 9, 2]). In contrast, this work attempts to closely integrate traditional or formal financial reasoning more closely into an intelligent system.

The design and configuration of the genotype was designed to be as flexible in allowing the system to define the numbers of rules and so on. For this reason, for instance, a large number of "possible" rules was specified ($O = 30$) that is never reached in actual chromosomes that specify the linguistic determinations of the rules because of setting a penalty parameter to penalize solutions containing rules which contribute no benefit.

There are, however, a number of design decisions which were taken fitting in with existing knowledge, such as using triangular membership functions rather than, say, Gaussian functions. Membership functions are a crucial part of the representation that define the mappings to assign meaning to input data. They map input observations of data to degrees of membership in fuzzy sets to describe properties of the linguistic variables. Suitable membership functions are designed depending on the specific characteristics of the linguistic variables as well as peculiar properties related to their use in optimization systems. Triangular membership functions are widely used primarily for the reasons described in [91]. Other common mappings include 'gaussian' [55] and 'trapezoidal' [39] membership functions. The functions are either predefined or determined in part or completely during an optimization process. A number of different techniques have been used for this task including statistical methods, heuristic approaches [6], and genetic and evolutionary algorithms [42, 82, 29, 99]. Adjusting membership functions during optimization is discussed in [42, 107].

Finally, on the topic of selection of input data, this chapter described a system with a specific set of (9) technical inputs. This set is extended in the subsequent chapters. Parameters such as the number of membership functions, inputs used, are made to be adapting in later chapters as well so as to, among other benefits, decrease the sensitivity of the approach to parameter settings and associated issues of “data mining” or fitting models to historic data such that generalization is poor.

Chapter 5

Initial Experimentation

This section contains a performance analysis of the system presented in the previous chapter. Section 5.1 provides details of the experimental setup that was used to produce the results used in the performance analysis including parameter settings and initial conditions.

In estimating the effectiveness of technical trading, focus is usually placed on the success of utilizing one specific type of rule in isolation to others. A moving average rule, for example, might be examined with varying moving average lengths and tested on a series of indices or stocks. However, a cursory examination of any of the currently popular technical trading books or websites that investors read regularly cite that one should never place complete faith in a single trading rule, and instead check for confirming signals from other indicators. A technical trader, therefore, is unlikely to trade using just one rule and would instead utilize a combination of chosen rules to determine when to buy or sell a stock. The problem from an academic point of view is that it becomes considerably difficult to test the success of such a strategy where there are an endless supply of combinations of rules that could be put together to determine a functional trading strategy. Nevertheless, by-and-large, technical traders do exactly this, choosing a certain set of rules to determine trades.

We use a very different approach which generates multiple trading rules as part of a fuzzy logic rule base. Kozas' (1992) work is often quoted in the finance literature as the basis for the development of genetic programming, which evolves specific computer generated formulae that can be utilized as a trading rule. This methodology, however, is a subset of a much larger body of optimization based on natural processes. This area of research has been evolving since the 1950s (see Fogel, 1998) however, it is only in the past decade or so that greater computer power that is readily available that these methods have become feasible problem solving techniques applied successfully in a wide range of applications in engineering, design and many others (see Chapter 3).

Another related issue is that traders will use a different form of the same trading rule over time. A simple case would be with, again, the moving average rule. A 20-day moving average rule might work well, relative to a 50-day rule, for a trader in one month but terribly in the next month. The moving average length may need adjustment. These matters are further complicated when examining rules that comprise more than one moving average series, or use a multiple number of parameters that can be varied to construct a particular indicator. Although it is quite easy to comment on the best parameter values to select ex post, knowing which values to use in the future is another issue entirely.

By using natural learning computation techniques we will test the hypothesis that technical trading rules can lead to profitable outcomes when used in a manner that is more akin to how actual practitioners trade utilizing these rules. The computational intelligence system develops buy signals based on combining various rules together into sets of rules that we refer to as rule bases. These rule bases, are dynamically updated every trading period for changes in the market and are utilized to determine the composition of a portfolio containing listed stocks that is re-balanced at regular monthly intervals. To minimize the problem of data-snooping the system also reduces the number of fixed parameters utilized within the system so that a large set of parameter specifications within the system are dynamically and adaptively linked to changing market conditions. The portfolio generated from the system, and comparison portfolios that utilize different objective functions for the rule bases are then examined using standard portfolio measurement tools. To determine relative performance we also compare the performance of these portfolios against Fama and French factor portfolios through the use of a stochastic dominance efficiency test.

These experiments presented here study the benefits of the approach from this perspective and to compare the performance of an evolving strategy to static methodologies from a search and also with well known trading rules from the financial literature. We also compare the algorithm performance of the evolutionary search optimization method with a more simple hill climbing heuristic. The analysis is divided into two parts: the first, Section 5.2, comprises an analysis using standard portfolio evaluation tools widely accepted by finance practitioners and researchers; the second, Section 5.3, consists of an evaluation using stochastic methods that do not make any assumptions about the characteristics of the return distributions. In the following discussion we refer to the portfolio generated using the computational intelligence system presented in this paper as the Evolutionary Algorithm (EA) portfolio.

5.1 Experimental Setup

This section provides parameters and initial set up of the evolutionary algorithm and associated system settings used to obtain the results given in the subsequent sections. Explanations of these parameters are given in Chapter 4.

5.1.1 Information set

The system was tested using historical data for stocks in the MSCI Europe index from 1990 to 2005. The MSCI Europe index represents the largest stocks, by market capitalization, which are traded across Europe. The MSCI Europe is in fact primarily composed from the individual country indices that MSCI creates and tracks. Although the constituent stocks that make up the index change over time, between 1990 and 2005 there were at least 700 active stocks that comprised the index at any point in time, with a total of 1241 represented over the whole period.

Two input files were used: one containing series for the trading volume of each stock, the other containing price data. The linguistic variables used are based on well known technical indicators used by real traders, and they were calculated solely using price and volume data. All stock data was adjusted for various company events that would alter the price of individual stocks. This would include, for example, share splits and the payment of dividends. All payments generated from a stock were assumed to have been reinvested back into the same stock. Also, share prices were converted to all be in the Euro. Where necessary, DataStream International synthetic Euro FX rates were utilized for currencies without a direct relationship with Euro or ERM prior to it becoming a physical currency.

The risk free rate of return used to calculate the alpha of stocks plus performance evaluation statistics provided in the next section are from the 3-month Euro deposit rate series that was taken from DataStream International.

A listing and brief description of the meaning of each linguistic variable is provided below with reference to a day t when the signal applies:

1. *Price Change*: the change in price over a 20 day period before day t,

2. *Single Moving Average Buy Signal*: the difference between the price at time t and a 20 day moving average at time t when the price is greater than the moving average,

3. *Single Moving Average Sell Signal*: the difference between the price at time t and a 20 day moving average at time t when the price is less than the moving average,

4. *Portfolio Value*: the value of the portfolio at time t,

5. *Double Moving Average Buy Signal*: the difference between a 10 day moving average at time t and a second moving average based on a longer time period (20 days) at time t when the first moving average is greater than the double moving average,

6. *Double Moving Average Sell Signal*: the difference between a 10 day moving average at time t and a second moving average based on a longer time period (20 days) at time t when the first moving average is less than the double moving average,

7. *On Balance Volume Indicator (OBV) Buy Signal*: the OBV indicator compares volume to price movements. A running indicator termed the OBV indicator is constructed such that volume is added if the closing price at time t of the indicator is higher than the previous closing price (at $t-1$), subtracted if it is lower and does not change if the closing price remains static. A buy signal is produced whose strength depends on the extent of divergence between the maximum price and the maximum OBV over a period from t to $t-20$.

8. *On Balance Volume Indicator (OBV) Sell Signal*: An OBV sell signal is produced whose strength depends on the extent of divergence between the minimum price and the minimum OBV over a period from t to $t-20$.

9. *Alpha*: an indicator based on the Capital Asset Pricing Model (see Section 5.2).

5.1.2 Parameters

The probabilities for applying the three operators described in 4.2 were as follows. For mutation there was probability 0.4, for uniform cross over 0.3 and for rule crossover 0.3. For the repair operator we used $p = 10\%$ in all experiments.

In Section 4.2 we mentioned the initial population can be seeded with predetermined rule bases. At the beginning of each optimization including for every window in the sliding window schema a single price momentum strategy rule base was inserted into the population, its phenotype was:

- If *Price Change* is *Extremely Low* then *rating* = 0.0
- If *Price Change* is *Very Low* then *rating* = 0.16
- If *Price Change* is *Low* then *rating* = 0.33
- If *Price Change* is *Medium* then *rating* = 0.5
- If *Price Change* is *High* then *rating* = 0.67
- If *Price Change* is *Very High* then *rating* = 0.83
- If *Price Change* is *Extremely High* then *rating* = 1.0

A sliding window methodology was used with a 120 day window with a 20 day window movement between periods. The real trading portfolio used to evaluate the results was generated using a rule base from the previous window. For the trading simulation (see Section 4.3.1) the parameters *buy best stocks percentage* was set to 10% and *sell worst stocks percentage* was also set to 10%. An additional constraint was also set that the maximum number of companies that the portfolio could take a position in at any one time be limited to 100 stocks.

The stopping condition for the evolutionary algorithm was such that the algorithm would run for each data window for a number of generations controlled by a parameter max steps without improvement (MSWI) which allows the has meaning that the algorithm continues iterating until MSWI iterations passed without a better rule base being found. In these experiments MSWI = 5000.

5.2 Standard Performance Analysis

In order to test and evaluate the performance of the EA portfolio, not only is a benchmark portfolio required, but also a comparison should be made with alternative strategies. A comparison with other traded funds would not necessarily be suitable, as the EA has been restricted to only utilize price and volume data. To determine the best buys in the market traded funds in the market are obviously able to also apply a wealth of company information ranging from cash flows, earnings and dividend behavior to name but a few. Therefore, we instead focus our main efforts into comparing the performance of the EA portfolio to two other more traditionally constructed portfolios that use the same information set available to the EA. The first of these portfolios is constructed from a price momentum strategy. Every 20 days the portfolio is re-balanced to hold the top 10% of stocks that are the best performing, in terms of returns over the previous 120 day period. There is sufficient academic research to indicate price-momentum strategies can outperform a passive index-tracking portfolio. See [53] for a recent discussion on the profitability of price momentum strategies and the potential reasons behind it. It is recognized that the strategy utilized in this paper is different from that discussed in the aforementioned paper in terms of length of holding period and ability to short sell. However, by constraining the price momentum design in this way, it will be utilizing the same dataset and trading constraints applied by the EA process. The results from this type of portfolio will therefore provide an indicator as to whether the EA portfolio does more than just replicate a momentum strategy.

The second portfolio is an alpha portfolio based on the single-factor regression model (the model relates a stocks excess return, $r_{i,t} - r_{f,t}$ to market return):

$$r_{i,t} - r_{f,t} = \alpha_i + \beta_i \left[r_{m,t} - r_{f,t} \right] + e_{i,t}, \tag{5.1}$$

where, each index i indicates a stock and each t refers to the i-th stock on a day; furthermore, e is an error term and $r_{i,t}$ is the stocks return on day t, $r_{f,t}$ is the risk free rate of return, $r_{m,t}$ is the market return. Theoretically, in an efficient market it would be possible to price stocks based solely on their risk components. Under this classical Capital Asset Pricing Model there is only one risk factor, that being the systematic risk of the stock (see Chapter 2). Therefore, excess returns of any stock, i, above the risk-free rate, r_f, can be fully explained by its level of systematic risk, β_i, and the market risk premium $(r_m - r_f)$. The alpha value of the stock, α_i, should be zero. If it is not and in fact there is a positive value then the stock is outperforming relative to its level of systematic risk and should be bought. The higher the alpha value, the better the stock is to purchase. An alpha value is calculated for each stock every 20 days using stock returns from the previous 60 days of trading data. Stocks with the highest alphas are bought and held.

We recognize that the above single-factor model is a relatively basic model of risk, and does not take into account more commonly used frameworks such as Fama and French's three-factor model [32], incorporating size and book to market value effects. One can also question the validity of calculating alphas over short periods of only 60 days, and the statistical significance of them. However, tests using various lengths of time to calculate alphas did not lead to radically different results. Moreover, the single-factor model explained above is congruent with forming a portfolio using only price information. It is also a subset of the information set utilized by the EA itself, and as such can provide some measure of relative performance to the EA portfolio from its ability to deviate away from standard price momentum and alpha-based strategies.

	MSCI Europe	EA	Alpha	Buy-and-Hold	Price Momentum	Hill Climber
Arithmetic Returns	187.25%	782.98%	258.72%	224.48%	175.94%	78.02%
Geometric Returns	8.61%	19.09%	10.78%	10.25%	8.46%	4.70%
Annualized Volatility	18.96%	18.07%	25.81%	19.69%	25.63%	16.98%
Sharpe Ratio	0.5307	1.0630	0.5251	0.5948	0.4452	0.2825
Jensen Alpha	NA	16.62%	9.16%	9.85%	7.93%	5.17%
Modified Alpha	NA	16.57%	9.09%	9.76%	7.86%	5.06%
Information Rank	NA	0.9524	0.3734	0.5066	0.3189	0.3061
Net Selectivity Measure	NA	9.62%	-0.01%	1.26%	-2.12%	-2.08%

Table 5.1: Standard Portfolio Performance Measures of testing results for daily return measures over the test period for each of the portfolios tested. All figures are for portfolios that were originally created on 16th November 1992 and held until 19th September 2005 using Euro as the base currency. The Sharpe ratio is calculated from annualized arithmetic returns. Excess returns are based on comparison with the 3-month Euro deposit rate. The quoted Alphas, Information Rank and Net Selectivity measures have been annualized.

Three further portfolios were also created. The first being a hypothetical MSCI Europe passive index. This essentially mimics the returns from the MSCI total return index itself and is set as the raw benchmark for all portfolios. The second is a buy-and-hold portfolio created by holding a selection of stocks based on optimizing the initial window, as discussed in Section 4.1. This will provide for a comparison of the EA performance against a static model. Finally, results from a hill climb optimization routine is also provided to compare the EA against another search-based optimization approach. The algorithm was initialized with a random rule-base of the same type as used by the EA and is based on the mutation operator described in Section 4.2, which enables the search to avoid being trapped in local optima. The solution is progressively improved through iterations. The algorithm is terminated when no improvement is found after 5,000 iterations.

At a first glance it is noticeable that the EA portfolio has performed exceedingly well when examined from an investors point of view who would have held the portfolio from inception until the end of the sample period. In fact, the EA provides an excess holding period return of 782.98%, this being more than four times the excess holding period return generated from an investor that had simply bought into a passive fund that tracked the market index (earning a return of 187.25%). To illustrate this, Figure 5.1 tracks the value of each portfolio for the 13 year holding period. Annualized excess returns for the EA were more than double (at 19.09%) to the market index. Interestingly, this higher return performance was not at the expense of higher risk, with annualized standard deviations below that of the MSCI index.

From a visual inspection of Figure 5.1 it is interesting to observe that the alpha and price momentum strategies seemed to perform quite well from 1992 to 2000 when for the most part the MSCI index followed an upward drift. The bearish market conditions thereafter did not help either portfolio perform as they did in the past. This is to be somewhat expected as the two strategies are more aligned for working with bull runs. There is a substantial body of research analyzing the potential reasons for the success of such simple strategies as that of a price momentum (for a summary, see [53]). However, it is also interesting to note that despite the change in market sentiment, the EA portfolio did not lose anywhere close to the same amount of money that the alpha and price momentum portfolios declined by, as it would seem that the adaptive trading rules utilized by the EA were able to evolve to the bearish phase in the financial market. This highlights well the importance of having an evolving rule-base to adapt to new market conditions. The Buy-and-Hold rule-base portfolio declines in value with the alpha and price momentum portfolios,

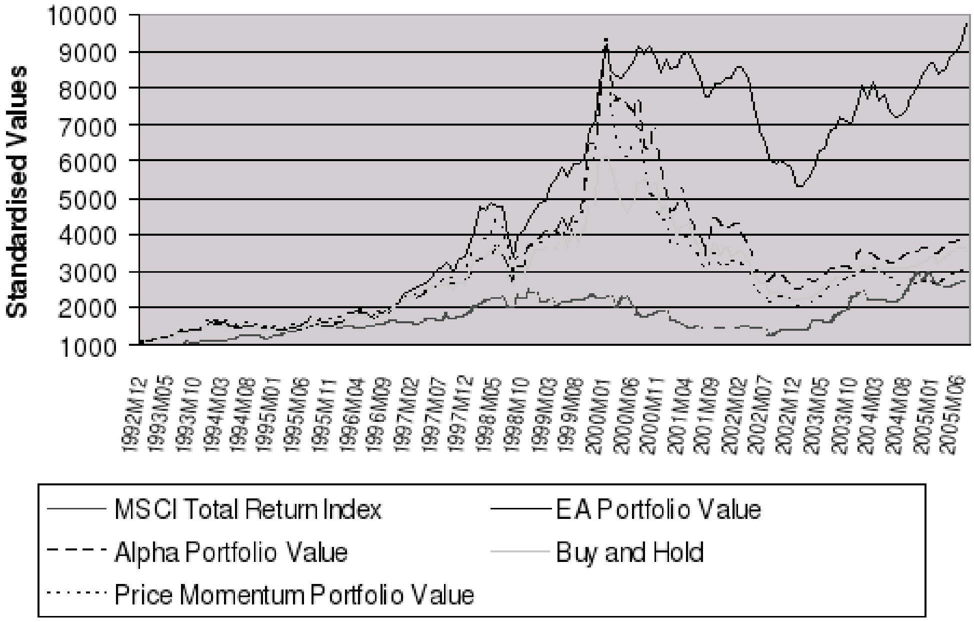


Figure 5.1: Testing results of portfolio values of a single portfolio for each of the approaches tested from from 1992 to 2005. Each starts at a value of 1000 on 16th November 1992.

suggesting further that the initial set of trading rules were suited for a bullish market and not suitable for bear runs. Interestingly, the hill climb approach also falls in value, suggesting the rule-bases were not as adaptive to changing market conditions as the EA method.

As one of the most popular and easily recognizable methods to compare portfolios is through their Sharpe ratios [98], Table 5.1 tabulates these results. The Sharpe measure is calculated as the returns of the portfolio, r_p, above the risk-free rate, r_f, divided by the portfolio standard deviation:

$$Sharpe = \frac{(r_p - r_f)}{\sigma_p}$$

As a measure of total risk adjusted return performance, only the EA and Buy-and-Hold portfolios were able to beat the market index. The slightly higher returns from the alpha portfolio did not sufficiently compensate for the far higher level of risk (a standard deviation of 25.81%). What is also of interest to note is the relatively high sharpe ratio for the EA portfolio at 1.063. Once total risk, as measured by the standard deviation of portfolio returns, is taken into account the EA portfolio stands out amongst all of the alternatives.

The next four measures are all based on the single-factor model and relate the performance of the portfolios to the benchmark, MSCI Europe index. The first two measures tabulate the portfolio alphas. These are similar to a stock's alpha, but relate to how much better the portfolio has performed relative to the systematic risk of the portfolio and performance of the benchmark index. All of the portfolios show some degree of over-performance, having positive alphas. However, only the alpha statistic from the EA portfolio was found to be significant at the 1% confidence level. Modified (see [63]) alpha values are also tabulated. These alpha values have been computed to take into account the fact that the returns series may not be normally distributed. [1] However, there is actually no significant difference in the figures presented. The robustness of these alpha values can also be measured through

[1] The modified alpha is calculated as:

$$B_p = \frac{Cov\left[r_p, -(1+r_m)^{-b}\right]}{Cov\left[r_m, -(1+r_m)^{-b}\right]},$$

where

$$b = \frac{ln(E\left[1+r_m\right]) - \ln(1+r_f)}{Var\left[\ln(1+r_m)\right]}.$$

the information performance rank that is presented. Sometimes also known as the appraisal ratio, it measures the portfolios average return in excess of the benchmark portfolio over the standard deviation of this excess return. Essentially, it evaluates the active stock-picking skills of the strategy, once unsystematic risk generated from the investment process is accounted for. As we are comparing each of our portfolio's with the MSCI Europe total return index, the information ratio is calculated as:

$$\text{Annualized Information Ratio} = \frac{\sqrt{T}\alpha}{\sigma_e},$$

where T is the period multiple to annualize the ratio and σ_e is the standard error of equation 5.1. Compared to other funds in the market, an appraisal ratio of 0.95 for the EA portfolio is indicative of a very strong and consistent performance. Grinold and Kahn, [41], have argued good information ratios should be between 0.5 and 1, with 1 being excellent. Goodwin [40] examined over 200 professional equity and fixed income managers over a ten year period and found that although the median information ratio was positive, it never exceeded 0.5. Of all the alternative portfolios, only the Buy-and-Hold portfolio comes close to beating the 0.5 value.

The final row in Table 5.1 presents the results from Fama's Net Selectivity measure [30]. It provides a slightly more refined method to analyze overall performance for an actively managed fund. Overall performance, measured as the excess returns of the portfolio over the risk-free rate, can be decomposed into the level of risk-taking behavior of the strategy and security selection skill. This security selection skill, or Selectivity, can be measured as a function of the actual return of the portfolio minus the return that the benchmark portfolio would earn if it had the same level of systematic risk. This selectivity value, however, can be broken down still further to calculate Net Selectivity. Given that a portfolio's strategy may not be limited to simply track the benchmark portfolio – which would be the case for our portfolios under examination – it is also necessary to take into account the fact that the portfolios are not fully diversified, relative to the chosen benchmark. In fact, for the EA and Buy-and-Hold portfolios the maximum number of stocks that it is allowed to have is restricted to 100, far less than the MSCI index. To account for this, net selectivity is the value of selectivity that the strategy adds to the portfolio minus the added return required to justify the loss of diversification from the portfolio moving away from the benchmark. This effectively means any returns that the portfolio earns above the risk free rate must be adjusted for *both* the returns that the benchmark portfolio would earn if it had the same level of systematic risk and the same level of total risk to the benchmark.

The net selectivity figures quoted will, by default, all be less than the alpha values previously examined. However, even when the differences in total risk are accounted for, the EA portfolio provides a very positive result. In fact, the only other portfolio to show a positive net selectivity figure is, again, from the Buy-and-Hold rule base.

	MSCI Europe Total Return Index	EA	Alpha	Buy-and-Hold	Price Momentum	Hill Climber
Average monthly return	0.8280%	1.6629%	1.2095%	1.0152%	1.0272%	0.5196%
Median monthly return	1.1648%	2.0782%	1.7063%	1.3757%	1.5490%	0.7591%
Largest positive monthly return	14.0260%	17.0171%	28.6675%	22.3981%	29.8286%	18.0645%
Largest negative monthly return	-12.8476	-17.0083	-19.5658	-16.1300	-24.3551	-25.2463%
Average monthly volatility	5.4607%	5.1996%	7.4242%	5.6528%	7.3720%	4.9016%

Table 5.2: Standard Performance Measures of testing results for daily returns for a test run of each of the portfolios tested. Monthly returns are calculated on a discrete basis as a percentage change from one day to the next.

Table 5.2 and 5.3 shows general distribution characteristics of the portfolios under examination. A Jarque-Bera test [10] shows that none of the constructed portfolios are normally distributed with the exception of the MSCI index. The EA portfolio shows evidence of negative skewness, implying from an investor's perspective that the majority of returns are generally above the mean, although large negative returns can be expected on an irregular basis. With the exception of the MSCI index all series demonstrate fat tails. In particular, the hill climb exhibits far

	MSCI Europe Total Return Index	EA	Alpha	Buy-and-Hold	Price Momentum	Hill Climber
Skewness	-0.0247	-0.1863	0.2633	-0.0322	0.0552	-0.667352
Kurtosis	3.0383	4.3629	5.3485	4.2690	5.8977	8.907055
Jarque-Bera	0.02491	12.7255[a]	36.9289[a]	10.2929[a]	53.6072[a]	226.1609[a]
Probability of a loss greater than 10% in any given month	4.58%	2.61%	11.76%	5.88%	16.99%	4.70%
Probability of a gain greater than 10% in any given month	3.92%	4.58%	7.84%	0.65%	1.31%	4.00%
Number of months before a negative monthly return	2.4	3.1	2.7	2.5	2.7	2.3

Table 5.3: Testing results statistics on portfolio returns for each of the approaches tested continued. The Jarque-Bera statistic is a chi-square distributed test for normality within the series. [a] signifies rejection of the null hypothesis of a normal distribution at the 1% significance level.

more excess kurtosis and skewness than the other portfolios. The excess kurtosis would lead to more regular, larger swings away from the mean in investor returns when compared to the EA portfolio. From an investors perspective, this is not particularly desirable and is investigated further in the stochastic dominance tests that are conducted.

One of the reasons for the shape of the distribution that has arisen from the EA strategy could be due to the specific fitness and penalty functions imposed upon the system. To investigate the tail ends of the returns distribution for the portfolios the table also reports some basic probability statistics. Specifically, the probability of experiencing a gain or loss greater than 10% in any given month. From these figures it is noteworthy that it is the EA portfolio that has the greatest chance of producing a monthly return in excess of 10%, and the smallest chance of producing undesirable negative returns greater than 10%. The probability of these occurring in any given month is 4.58% and 2.61%, respectively. These results may be indicative of the penalty function correctly discarding the choice of stocks that are more likely to experience a large decline. Although the penalty function can be viewed as a means to ensure the fitness function is geared more closely towards being a risk-adjusted return, it is not the same as employing a Sharpe ratio or other standard deviation measure. The difference being that the penalty function only penalizes for large downside risk, rather than both up and downside risk.

The table also provides a simple measure of how often a negative monthly return can be expected for an investor holding the relevant portfolios. Once again, it is the EA portfolio that performs the best out of the alternative strategies, experiencing a negative return only once every 3 months.

The results presented here focus on daily returns of portfolios managed by the system and combine many separate runs over a sliding window where the portfolio is updated with a new rule set from a separate optimization procedure each month. An additional result set focusing on final results and combinations of many distinct runs is given in Chapter 8.

To cater for the fact that the returns distributions are non-normal, the following Section, 5.3, evaluates the relative performance of each of the constructed portfolios with the MSCI index using non-parametric, distribution free stochastic dominance tests. These will go someway to deal with the fact that upside and downside movements in the above portfolios are not symmetric.

5.3 Stochastic Dominance Analysis

The concept of stochastic dominance (SD) gives a systematic framework to analyze investment choices under uncertainty, utilizing only some general assumptions on an investor's utility function. The attractiveness of the method is therefore on it not requiring any knowledge of the statistical distribution of the investment alternatives. It provides a statistical comparison between portfolios using the whole distribution, rather than just point estimates.

In CAPM analysis, the efficiency criterion uses only the mean and variance of the returns, based on the underlying assumption that returns are distributed normally. As discussed in Section 5.2, none of the return distributions of our portfolios are normally distributed. SD efficiency criteria do not require this distributional assumption. The three most general SD efficiency criteria are:

1. First degree stochastic dominance (FSD) rule. This is the smallest efficiency criterion which produces the smallest possible efficient set for all rational investors — individuals with an increasing utility function.

2. Second degree stochastic dominance (SSD) rule. This is the smallest efficiency criterion which produces the smallest possible efficient set for all risk averse investors.

3. Third degree stochastic dominance (TSD) rule. This is the smallest efficiency criterion which produces the smallest possible efficient set for all rational investors, who are risk-averse and have decreasing absolute risk aversion.

To answer the question of whether the EA portfolio is a superior investment choice for any of the above three types of investors, we need to test whether the return distribution generated by the EA rules dominates the alternative strategies. This is achieved by conducting pair-wise tests of stochastic dominance:

$$\begin{aligned} H_0^j &: \text{ } G \text{ dominates } F \text{ stochastically at order } j, \\ H_1^j &: \text{ } G \text{ does not dominate } F \\ &\quad\ \text{stochastically at order } j, \end{aligned}$$

where G and F are two cumulative return distributions generated from two different technical strategies. The hypotheses can be written compactly as

$$\begin{aligned} H_0^j &: J_j(z;G) \leq J_j(z;F) \\ &\qquad \text{for all } z \in [0,\bar{z}], \\ H_1^j &: J_j(z;G) > J_j(z;F) \\ &\qquad \text{for some } z \in [0,\bar{z}], \end{aligned}$$

where $[0,\bar{z}]$ is the common domain of F and G and $J_j(.;G)$ is the function that integrates the function G to order $j-1$ so that, for example:

$$\begin{aligned} J_1(z;G) &= G(z), \\ J_2(z;G) &= \int_0^z G(t)dt \\ &= \int_0^z J_1(t;G)dt, \\ J_3(z;G) &= \int_0^z \int_0^t G(s)dsdt \\ &= \int_0^z J_2(t;G)dt, \end{aligned}$$

and so on.

Recently, Barret and Donald [8] proposed a set of Kolmogorov-Smirnov type tests (KS tests) for SD of any order. The KS tests compare two distributions at all points in the domain range, therefore having the potential to be consistent tests of the full restrictions implied by SD. The tests also allow for different sample sizes, and the p-values are generated via a variety of simulation and bootstrap methods. Table 5.4 and Table 5.5 report the p-values for various tests of pair-wise dominance between all five portfolios under consideration. In Table 5.4, the p-values are calculated via 2 different Monte Carlo simulation methods, whereas in Table 5.5 the p-values are calculated using 3 different bootstrapping procedures. The null hypothesis of G dominance over F is rejected at 95% level of confidence if the p-value is smaller than 0.05.

Table 5.4: KS test for SD - Monte Carlo simulation
This table reports the p-values of tests for the dominance of portfolio G over portfolio F

Panel 1. P-values calculated by Monte Carlo simulation method 1

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.01	0.01	0.04	0.01	0.71	-	0.00	0.00	0.01	0.00	0.67
PM	0.52	-	0.45	0.39	0.19	0.23	0.69	-	0.64	0.70	0.47	0.02	0.65	-	0.66	0.65	0.67	0.09
Al	0.52	0.59	-	0.45	0.45	0.12	0.69	0.43	-	0.55	0.30	0.03	0.64	0.31	-	0.66	0.67	0.14
BH	0.94	0.45	0.59	-	0.23	0.19	0.65	0.12	0.16	-	0.29	0.16	0.58	0.04	0.12	-	0.26	0.41
Idx	0.9	0.59	0.59	0.85	-	0.02	0.52	0.09	0.15	0.46	-	0.11	0.49	0.11	0.18	0.56	-	0.38
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.21	0.11	0.11	0.24	-	0.00	0.59	0.52	0.16	0.36	-

Panel 2. P-values calculated by Monte Carlo simulation method 2

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.01	0.01	0.04	0.01	0.69	-	0.00	0.00	0.01	0.00	0.65
PM	0.52	-	0.45	0.39	0.19	0.23	0.72	-	0.65	0.72	0.47	0.02	0.67	-	0.66	0.67	0.65	0.11
Al	0.52	0.59	-	0.45	0.45	0.12	0.73	0.44	-	0.57	0.30	0.03	0.69	0.33	-	0.68	0.66	0.13
BH	0.94	0.45	0.59	-	0.23	0.19	0.66	0.13	0.17	-	0.30	0.15	0.58	0.05	0.13	-	0.28	0.41
Idx	0.90	0.59	0.59	0.85	-	0.02	0.57	0.11	0.17	0.49	-	0.08	0.52	0.14	0.21	0.49	-	0.36
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.22	0.12	0.10	0.29	-	0.00	0.63	0.55	0.15	0.41	-

Legend — EA: evolutionary algorithm; PM: price momentum; Al: alpha; BH: Buy-and-Hold;
Idx: msci total return index; Hc: hill climbing

Table 5.5: KS test for SD - Bootstrap methods

This table reports the p-values of tests for the dominance of portfolio G over portfolio F

Panel 1. P-values calculated by bootstrap method 1

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.00	0.00	0.00	0.00	0.20	-	0.00	0.00	0.00	0.00	0.20
PM	0.52	-	0.45	0.39	0.19	0.23	0.60	-	0.40	0.60	0.40	0.00	0.20	-	0.20	0.20	0.20	0.00
Al	0.52	0.59	-	0.45	0.45	0.12	0.80	0.60	-	0.80	0.60	0.00	0.80	0.20	-	0.80	0.80	0.00
BH	0.94	0.45	0.59	-	0.23	0.19	0.40	0.00	0.00	-	0.00	0.00	0.40	0.00	0.00	-	0.00	0.00
Idx	0.90	0.59	0.59	0.85	-	0.02	1.00	0.40	0.40	0.80	-	0.00	0.80	0.40	0.40	0.80	-	0.00
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.00	0.20	0.00	0.60	-	0.00	0.20	0.80	0.00	0.40	-

Panel 2. P-values calculated by bootstrap method 2

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.20	0.20	0.20	0.00	0.80	-	0.20	0.20	0.20	0.00	0.80
PM	0.52	-	0.45	0.39	0.19	0.23	0.80	-	0.80	0.80	0.80	0.20	0.80	-	0.80	0.80	0.80	0.40
Al	0.52	0.59	-	0.45	0.45	0.12	0.80	0.40	-	0.40	0.60	0.20	0.80	0.40	-	0.80	0.80	0.20
BH	0.94	0.45	0.59	-	0.23	0.19	0.80	0.20	0.20	-	0.60	0.20	0.80	0.20	0.20	-	0.60	0.80
Idx	0.90	0.59	0.59	0.85	-	0.02	0.80	0.20	0.20	0.60	-	0.40	0.80	0.20	0.20	0.40	-	0.60
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.40	0.00	0.20	0.00	-	0.00	0.60	0.60	0.20	0.20	-

Table 5.6: KS test for SD - Bootstrap methods (continued)

Panel 3. P-values calculated by bootstrap method 3

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.20	0.20	0.20	0.00	0.80	-	0.40	0.40	0.00	0.00	0.80
PM	0.52	-	0.45	0.39	0.19	0.23	1.00	-	1.00	1.00	0.80	0.60	0.80	-	1.00	0.80	0.80	0.60
Al	0.52	0.59	-	0.45	0.45	0.12	1.00	0.80	-	1.00	0.60	0.40	1.00	0.80	-	1.00	1.00	0.60
BH	0.94	0.45	0.59	-	0.23	0.19	0.80	0.60	0.60	-	0.80	0.60	0.80	0.60	0.60	-	0.80	0.80
Idx	0.90	0.59	0.59	0.85	-	0.02	0.80	0.80	0.80	0.80	-	0.20	0.80	0.80	0.80	0.80	-	0.60
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.20	0.40	0.40	0.80	-	0.20	0.80	0.80	0.60	0.80	-

Legend — EA: evolutionary algorithm; PM: price momentum; Al: alpha; BH: Buy-and-Hold;
Idx: msci total return index; Hc: hill climbing

We ran the tests of SD in both ways for each pair of portfolios. Portfolio A is said to be concluded as dominant over portfolio B if (1) the hypothesis that A dominates B is not rejected and (2) the hypothesis that B dominates A is rejected. It can be seen from Table 5.4 and Table 5.5 that there is no clear dominance patten among the five portfolios: Hill Climbing, MSCI Index, Buy-and-Hold, Price Momentum, and Alpha strategies. However, the EA portfolio is found to dominate the Hill Climbing in all orders, and dominate the other four portfolios in the second and third orders, implying that all risk-averse investors will favor the EA portfolio compared to the others.

One important assumption underlying the KS tests is the independence of the two samples coming from the two return distributions to be compared. In our case, even though the EA procedures do have some links with other portfolio generation rules, the correlation is between the return distributions themselves, rather than between the samples generated. However, we perform an additional SD test, as proposed by [24], which allows for interdependency between the samples tested. The test is basically a Maximal-T test, which compares two return distributions at a fixed number of points only. Conservative p-values (based on the widely applicable conservative critical values) and simulated p-values are reported in Table 5.7.

Table 5.7: Maximal T test for SD

This table reports the p-values of tests for the dominance of portfolio G over portfolio F

Panel 1. 10-point evaluation, simulated p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.05	0.14	0.50	0.05	0.29	-	0.01	0.02	0.15	0.07	0.70	-	0.02	0.02	0.17	0.13	0.82
PM	0.64	-	0.67	0.49	0.27	0.14	0.95	-	0.66	0.71	0.60	0.04	0.99	-	0.71	0.87	0.81	0.09
Al	0.34	0.34	-	0.34	0.34	0.14	0.92	0.52	-	0.65	0.51	0.05	0.98	0.44	-	0.83	0.75	0.13
BH	0.94	0.80	0.63	-	0.44	0.30	0.96	0.05	0.09	-	0.49	0.28	0.81	0.07	0.06	-	0.54	0.53
Idx	0.74	0.42	0.42	0.74	-	0.19	0.89	0.04	0.04	0.46	-	0.20	0.64	0.03	0.02	0.39	-	0.53
Hc	0.00	0.00	0.01	0.01	0.03	-	0.05	0.44	0.37	0.36	0.52	-	0.12	0.74	0.68	0.44	0.34	-

Panel 2. 10-point evaluation, conservative p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.09	0.21	0.42	0.09	0.50	-	0.09	0.11	0.19	0.33	1.00	-	0.13	0.15	0.20	0.53	1.00
PM	0.90	-	0.91	0.77	0.46	0.21	1.00	-	1.00	1.00	1.00	0.21	1.00	-	1.00	1.00	1.00	0.47
Al	0.58	0.58	-	0.58	0.58	0.21	1.00	0.99	-	1.00	0.99	0.23	1.00	0.97	-	1.00	1.00	0.57
BH	1.00	0.99	0.89	-	0.70	0.50	1.00	0.28	0.41	-	0.98	0.81	1.00	0.33	0.31	-	0.99	0.99
Idx	0.95	0.69	0.69	0.95	-	0.32	1.00	0.16	0.21	0.97	-	0.66	1.00	0.19	0.16	0.96	-	0.99
Hc	0.00	0.00	0.01	0.01	0.05	-	0.26	0.98	0.92	0.91	0.99	-	0.53	1.00	1.00	0.97	0.94	-

Table 5.8: Maximal T test for SD (continued)

Panel 3. 5-point evaluation, simulated p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.13	0.12	0.69	0.04	0.21	-	0.01	0.02	0.63	0.06	0.71	-	0.02	0.02	0.72	0.10	0.77
PM	0.63	-	0.50	0.49	0.26	0.21	0.93	-	0.63	0.68	0.56	0.04	0.98	-	0.67	0.83	0.76	0.08
Al	0.49	0.54	-	0.49	0.36	0.12	0.89	0.52	-	0.61	0.48	0.05	0.97	0.45	-	0.77	0.70	0.10
BH	0.84	0.80	0.60	-	0.35	0.21	0.93	0.12	0.19	-	0.45	0.25	0.87	0.05	0.06	-	0.52	0.50
Idx	0.85	0.89	0.95	0.75	-	0.12	0.95	0.06	0.11	0.50	-	0.17	0.72	0.02	0.02	0.39	-	0.51
Hc	0.00	0.00	0.01	0.01	0.05	-	0.26	0.98	0.92	0.91	0.99	-	0.53	1.00	1.00	0.97	0.94	-

Table 5.9: Maximal T test for SD - (continued)

Panel 4. 5-point evaluation, conservative p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.15	0.15	0.67	0.04	0.26	-	0.05	0.07	0.39	0.20	0.99	-	0.07	0.08	0.50	0.30	1.00
PM	0.81	-	0.67	0.66	0.34	0.26	1.00	-	0.97	0.98	0.93	0.14	1.00	-	1.00	1.00	1.00	0.25
Al	0.66	0.74	-	0.66	0.48	0.15	1.00	0.92	-	0.95	0.86	0.16	1.00	0.89	-	1.00	0.99	0.33
BH	0.97	0.95	0.80	-	0.48	0.26	1.00	0.31	0.47	-	0.85	0.62	1.00	0.17	0.21	-	0.93	0.93
Idx	0.97	0.98	1.00	0.91	-	0.15	1.00	0.20	0.29	0.88	-	0.43	0.99	0.10	0.10	0.81	-	0.93
Hc	0.00	0.01	0.01	0.01	0.05	-	0.04	0.41	0.33	0.31	0.48	-	0.09	0.69	0.63	0.45	0.41	

Legend — EA: evolutionary algorithm; PM: price momentum; Al: alpha; BH: Buy-and-Hold; Idx: msci total return index; Hc: hill climbing

The Maximal-T test results confirm the previous finding that there is no clear dominance pattern amongst the Price Momentum, Alpha, Buy-and-Hold and Hill Climbing portfolios. However, the tests give some support to the hypothesis that the Price Momentum and Alpha portfolios outperform the MSCI Index in the third order, ie. those investors who are risk averse and have decreasing absolute risk aversion will not choose to invest in the Index portfolio.

Similar to the KS tests, the EA portfolio is still found to be the best performing one. It dominates both the Price Momentum and the Alpha strategies in the second and third orders, and therefore is still the preferred choice for risk-averse investors.

Chapter 6

Adaptation

In this chapter we discuss mechanisms for adapting the evolving fuzzy rule base system discussed in Chapter 4. The method discussed here was presented in [38]. Figure 6.1 shows a summary of the components of the portfolio management process that is implemented and a hierarchy of relationships between them, including: the data loading module; the abstract models used to represent real trading; the prediction module; and a decoder that interprets a fuzzy rule base solution to imply decisions. The design allows for updating the strategy from feedback from performance in portfolio management. Let us now briefly describe the main components, where further detail of the basic operation fundamental elements of the design are given in Chapter 4. Here we focus on extensions.

Using input files containing price, volume, index and interest rate data in comma separated value file format, a database containing basic market data as well as derived data in the form of technical indicator values is constructed. Using this database we construct a data model containing abstract representations and models that are used to build training data windows for the optimizer and the trading model. The trading model is the universe of all aspects of the portfolio management task that are considered: it contains price and volume data from the data model, signal time series that are used as inputs to the fuzzy rules and also settings for parameters that introduce realism such as transaction costs and interest rate values. The trading model is an abstraction of a real trading environment that enables the prediction module to produce a forecasting model that takes into account elements of realism through the evaluation in simulation.

A forecasting model is a rule base and some parameters that describe how it should be applied that is found by an evolutionary search process with respect to a fitness evaluation process. The evaluation process involves simulated trading on fixed periods of recent historical data, a trading methodology and the simulacrum of real trading defined by parameters of the trading model. Each rule base specifies a particular pattern in a historical data window that resulted in the best training performance. A rule base enables the merit of stocks to be compared relative to the extent that the input data vector for the stock matches the pattern specified by the rule base. The fuzzy characteristics implement a practical definition of *extent*. This relative comparison of stocks results in a ranking of all feasible stocks which is then interpreted by the decoder module to imply recommendations for portfolio transactions. Section 6.1 describes the processes and models used by the prediction module.

The process depicted in Figure 6.1 is repeated each time the system conducts portfolio operations at set monthly intervals, it is at these points that the portfolio transactions take place. The feedback loops effect subsequent management operations by influencing the data window length or parameters used in the prediction module depending on the systems performance over time. An important distinction between the various adaptive mechanisms that are discussed is those which involve a reaction to the market (operating between optimization runs) and those that enable the optimization algorithm to self adjust within each optimization.

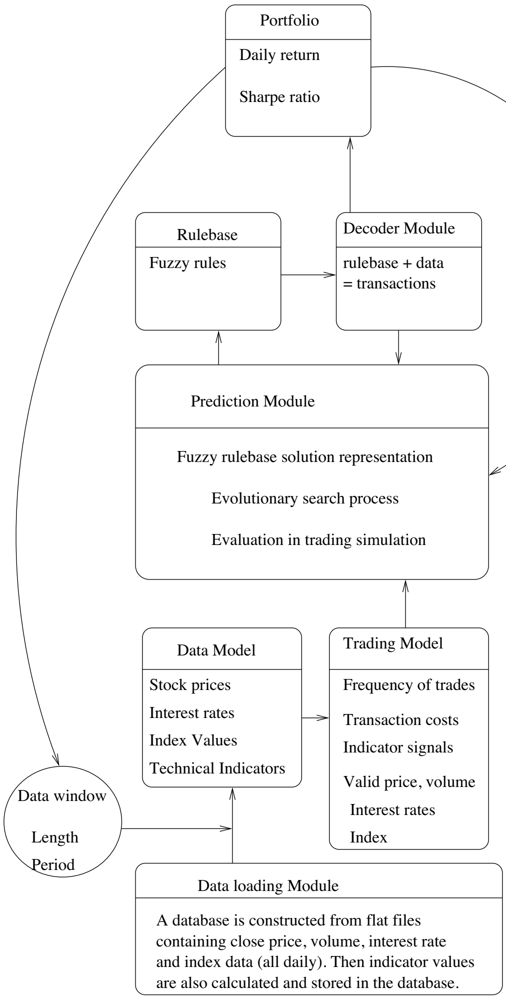


Figure 6.1: An overview of the system divided into components and the process that takes place each time the system conducts a management cycle on a real portfolio. Two feedback loops are in effect such that the risk and return performance of the managed portfolio over time is used to adjust the window length and parameters of the evolutionary search process (currently only a repair operator that maintains similarity between populations across training windows).

6.1 Search Process

The prediction module produces rule base solutions that lead to portfolio management decisions that satisfy the objectives. The rule base model is trained using an evolutionary search process. The process is shown in Figure 6.2. As depicted a repair operator is used to adjust the freedom of the search to explore new areas of the search space. The mechanism to compare or evaluate rule bases in the search is a trading simulation of the way rule bases are applied to managing the real portfolio using an adapting data window for training. The evaluation procedure is depicted in Figure 6.3. It involves simulated trading from which detailed financial statistics on performance are calculated from applying the rule base in historical data, this information is then utilized to evaluate the rule base performance and compare solutions.

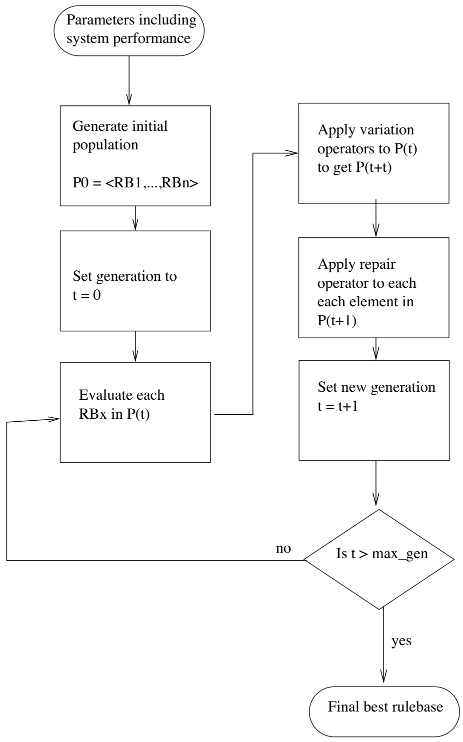


Figure 6.2: A flow chart describing the evolutionary algorithm (Figure 6.3 provides further explanation of the sub procedure for the evaluation in simulation. Selection was by tournament (size 2), elitism was not used).

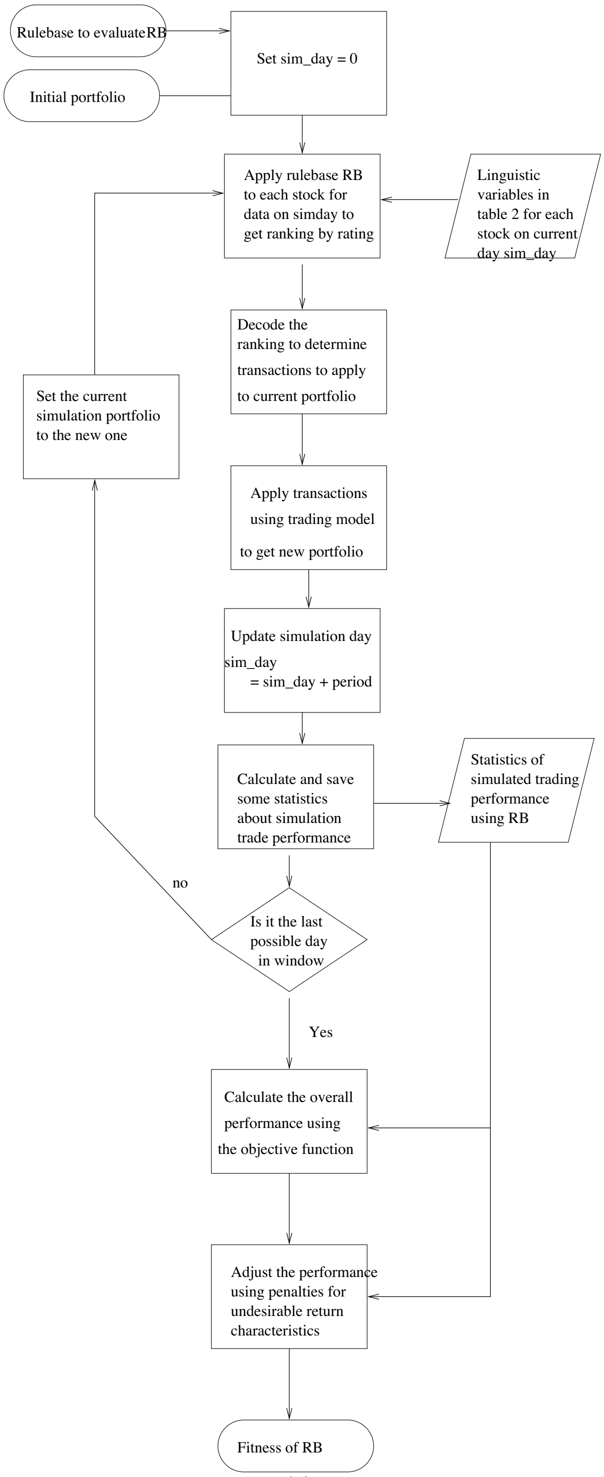


Figure 6.3: The evaluation process. Additional statistics are gathered relating to daily return performance compared with the system in Chapter 4.

6.2 Adapting the Algorithm

This section describes mechanisms introduced to allow the algorithm to adapt during the search. We consider the operators,

6.2.1 Adapting operator probabilities

This section discusses adaptation of genetic operators. The section 6.2.1 describes the adaptation of the search during a run. Section 6.3.1 discusses of adaptation of a repair operator used to control the way information remembered from previous searches is used to influence new searches depending on portfolio performance feedback relative to a benchmark.

The probability of applying operators to obtain the next generation is adjusted using an accumulation of statistics of their success in producing better offspring [56]. All the operators used act on genotype rule bases ρ that are specified internally by 4 arrays I (an integer array of input membership functions), O (a floating point array of outputs), UI (boolean values for inputs used in each rule, and UR (boolean switches indicating if rules in the genotype are used in the chromosome).

Each of the mutation and crossover operators provided in Tables 6.1 and 6.2 start the EA run with an equal probability of being applied to produce the next generation. In the implementation a the possibility of an operator being selected is increased relative to the others when that operator results produces an improved solution. Improved offspring produced by each operator (has higher fitness than both parents) are counted and every the probabilities are calibrated in proportion to the frequency of improvements.

For example in a system with 3 operators A, B and C if, over 100 generations, operator A results in 100 improved solutions, operator B results in 5 improved solutions and operator C in 50 (the remainder of generations did not find improvements) the probabilities for each operator would be: $P_A = 100/(100 + 5 + 50) = 100/155$, $P_B = 5/155$, $P_C = 50/155$.

6.2.2 Linguistic Variables and Definitions

The rule inputs, termed linguistic variables, discussed here are based on technical indicators used by finance practitioners that are calculated using close price and

Mutation Operators		
Type	Description	Output (new ρ)
Mutation	μ-SMALL	
	$\rho \mapsto \rho$	selects an element with equal probability from UR,I,UI or O and increments or decrements with equal probability by 1 appropriate unit (e.g. if boolean goes to false, if integer add or subtract 1).
Mutation	μ-LARGE	
	$\rho \mapsto \rho$	selects an element with equal probability from UR,I,UI or O and sets to a new legal value chosen with equal probability.
Mutation	μ-SMART	
	$\rho \mapsto \rho$	selects an element with equal probability from I or O such that UR,UI are true and increments or decrements with equal probability by 1
Mutation	μ-MF	
	$\rho \mapsto \rho$	selects an element with equal probability from $noMF$ and increments or decrements with equal probability by 1
Mutation	μ-LV	
	$\rho \mapsto \rho$	selects an element with equal probability from the set a, b, c, d increments or decrements with equal probability by 1

Table 6.1: The mutation operators.

Crossover Operators		
Type	Description	Output (new ρ)
Crossover	γ-UX	
	$\rho \times \rho \mapsto \rho$	uniformly select elements from UR,I,UI or O with equal probability to construct a new rule base from two parents
Crossover	γ-RULE	
	$\rho \times \rho \mapsto \rho$	swap rows from UR,I,UI or O to construct a new rule base with whole rules from two parents

Table 6.2: The crossover operators.

Repair Operators		
Type	Description	Output (new ρ)
Repair	r-LEGAL	
	$\rho \mapsto \rho$	genes in from ρ are changed until there are no illegal values (e.g integers in I less than 0 or greater than the number of membership functions for the corresponding variable)
Repair	r-FIXED	
	$\rho \times \rho \mapsto \rho$	genes from I and O in ρ_1 where UR and UI are true in ρ_2 are overwritten by corresponding values from ρ_2

Table 6.3: The repair operators.

volume data. In this section we discuss an extended set of variables whose meanings are calculated adaptively depending on some significant variables added to the genotype.

All the linguistic variables discussed here are calculated using daily data of stock closing price and volume. The selection of possible rule inputs is designed to enable the system to generate entry and exit signals for trades by considering indicators popular among finance practitioners and represented in financial academic literature [14]. They are based on different classes of technical indicators including, these different classes are used to forecast different categories of events in stock prices by technical analysts in practice.

The inputs are able to be divided into the following categories: moving average, momentum, oscillation, and breakout indicators and also indicators based on volume or price and volume rather than only price. Moving averages are often used to identify trends and to smooth out fluctuations due to daily or short, unsustained changes, depending on the period to calculate the average. New trends are identified when a moving average series crosses the price, or a shorter period average crosses a longer average. Oscillating indicators are used to identify cyclic patterns in price movements by compressing observations into a range, possibly giving more weight to recent points such as %R, and then generating buy or sell signals appropriately when extremes in the range are reached. Breakout indicators, as suggested by their name, are designed to catch significant changes in price direction at an early stage, for example a movement well outside the standard deviation of the mean historic returns is an indication that an unusual trend is emerging as opposed to a cyclic occurance. Volume data is an important input component and an indicator of market sentiment with links to behavioural aspects of market activity. In general, a market is considered strong by technical analysts if price and volume are both increasing.

Complex patterns in input data for particular stocks and the extent that new observed data matches these patterns for are modeled using a fuzzy rule base solution representation. A fuzzy rule base is a set of fuzzy rules, an example of rule base that could be represented by the system could be:

- If Price Oscillator is Very High then rating = 0.8
- If Single Moving Average Signal (a) is High and Double Moving Average (b,c) is Very High then rating = 0.7
- If RSI is Low and DMI is Very High and MFI is Extremely High then rating = 0.5

where a, b are integers that represent moving average lengths that define properties of the inputs such as moving average lengths and the period used to calculate the oscillator. The inputs, Price Oscillator, DMI, etc, are described by fuzzy membership sets such as Very High and Low etc. The input part of the rule is augmented by the variables a, b, c and d: these are integer variables that provide an additional level of rule optimization. Finally, an output rating is specified for each rule with respect to the particular inputs.

A rule base such as above is evaluated with respect to data observations to measure the extent that every rule is fulfilled given an input data vector. The output is a single floating value between 0 and 1. Each fuzzy rule is a series of conjunctions specifying membership levels of triangular membership functions that specify fuzzy sets extremely low to extremely high — there are a variable number of sets for each linguistic variable. The membership function definitions are tailored to the application domain. Some of the technical indicators used as rule inputs have opposite meanings for values that are either positive or negative. For these variables a central zero category divides the classifications around a central zero point. For example for the PPO indicator a value less than 0.5 is a negative signal but a value greater than 0.5 is a buy signal. These meanings are maintained in the rule bases. Figure 6.4 illustrates an actual set of membership function produced by the system for the PPO linguistic variable.

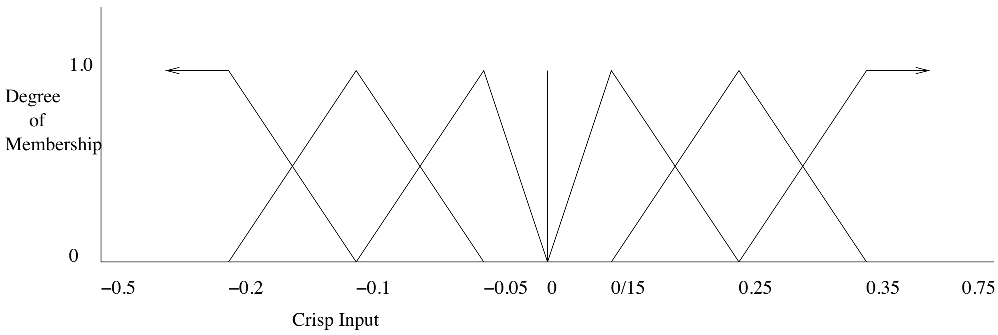


Figure 6.4: Example of the membership functions for a linguistic variable rule input: the figure shows the membership functions for the price oscillator (PPO) linguistic variable. In this case there are 7 membership sets: Extremely Low (EL), Very Low (VL), Zero (0), High (H), Very High (VH), Extremely High (EH). The figure is not to scale.

The set of input types includes: moving average indicators (Table 6.2.2), oscillation indicators (Table 6.2.2), breakout indicators (Table 6.2.2), momentum indica-

Name	Formula	Restrictions
Price Change 1	$\ln\left(\frac{p_t}{p_{t-\delta}}\right)$	$\delta = 20$
Price Change 2		$\delta = 50$
Price Change 3		$\delta = 100$

Table 6.4: Price momentum indicators.

Name	Formula	Restrictions
SMA Buy	$\frac{p_t}{ma_t}$	$len_{ma} \in \{10, 20, 30\}$
SMA Sell	$\frac{ma_t}{p_t}$	$len_{ma} \in \{10, 20, 30\}$
DMA Buy 1	$\frac{ma1_t}{ma2_t}$	$len_{ma2} \in \{10, 20, 30\}$ $len_{ma2} \in \{40, 50, 60\}$
DMA Buy 2		$len_{ma1} \in \{60, 70, \ldots, 120\}$ $len_{ma2} \in \{130, 140, \ldots, 240\}$
DMA Buy 3		$len_{ma1} \in \{60, 70, \ldots, 120\}$ $len_{ma2} \in \{130, 140, \ldots, 240\}$
DMA Sell 1	$\frac{ma2_t}{ma1_t}$	$len_{ma2} \in \{10, 20, 30\}$ $len_{ma2} \in \{40, 50, 60\}$
DMA Sell 2		$len_{ma1} \in \{60, 70, \ldots, 120\}$ $len_{ma2} \in \{130, 140, \ldots, 240\}$
DMA Sell 3		$len_{ma1} \in \{60, 70, \ldots, 120\}$ $len_{ma2} \in \{130, 140, \ldots, 240\}$

Table 6.5: Price moving average indicators. The abbreviations have the following meanings: SMA – single moving average; DMA – double moving average.

tors (Table 6.2.2), volume indicators (Table 6.2.2). Table 6.9 lists some conditions that must be satisfied for the calculations given in these tables to be defined. The abbreviations used in these tables have the following expansions: SMA, single moving average; DMA, double moving average; PPO, price percentage oscillator; OBV, on balance volume indicator; SD, standard deviation; RSI, relative strength index; MFI, money flow index; Bol, Bollinger band; Vol. DMA, volume double moving average; PVO, percentage volume oscillator; DMI, directional movement index; %R, percent R. Table 6.9 lists conditions that must be met for the corresponding linguistic variables to be defined and be used.

Adding additional elements to the genotype allows for evolution of the linguistic variable specification in parallel with the procedure to optimize the rest of the rule-

Name	Formula	Restrictions
PPO 1	$\frac{ma1_t - ma2_t}{ma1_t} \times 100$	$len_{ma2} \in \{10, 20, 30\}$ $len_{ma2} \in \{40, 50, 60\}$
PPO 2		$len_{ma1} \in \{60, 70, \ldots, 120\}$
PPO 3		$len_{ma2} \in \{130, 140, \ldots, 240\}$ $len_{ma1} \in \{60, 70, \ldots, 120\}$ $len_{ma2} \in \{130, 140, \ldots, 240\}$
DMI	see [106]	
%R	$\%R = \frac{p_t - min[p_{t-1}, \ldots, pt-10]}{max[p_{t-1}, \ldots, pt-10] - min[p_{t-1}, \ldots, pt-10]}$	
RSI	$RSI = 100 - \frac{100}{1+RS}$	$RS = \frac{totalgains \div n}{totallosses \div n}$
MFI	$MFI = 100 - \frac{100}{1+MR}$	$MR = \sum \frac{MF^+}{MF^-}$ $MF^+ = p_i \times v_t, wherep_i > p_{i-1}$, and $MF^- = p_i \times v_t, wherep_i < p_{i-1}$

Table 6.6: Price oscilator indicators. The following abbreviations were used: PPO – percentage price oscillator; DMI – directional movement indicator; RSI – relative strength index; MFI – money flow index.

Name	Formula	Restrictions
Vol. DMA Buy 1	$\frac{vma1_t}{vma2_t}$	$len_{vma1} = 5, len_{vma2} = 20$
Vol. DMA Buy 2		$len_{vma1} = 20, len_{vma2} = 100$
Vol. DMA Sell 1	$\frac{vma2_t}{vma1_t}$	$len_{vma1} = 5, len_{vma2} = 20$
Vol. DMA Sell 2		$len_{vma1} = 20, len_{vma1} = 100$
OBV Buy	$\frac{(p_t - \max[p_{t-1}, \ldots p_{t-n}])}{p_t} + \frac{(\max[obv_{t-1}, \ldots obv_{t-n}] - obv_t)}{obv_t}$	
OBV Sell	$\frac{(\min[p_{t-1}, \ldots, p_{t-n}] - p_t)}{p_t} + \frac{(obv_t - \min[obv_{t-1}, \ldots, obv_{t-n}])}{obv_t}$	
PVO 1	$\frac{ma1_t - ma2_t}{SMt} \times 100$	$len_{ma1} = 5, len_{ma1} = 20$
PVO 2		$len_{ma1} = 20, len_{ma1} = 100$

Table 6.7: Volume indicators. Abbreviations used have meanings: Vol: volume; DMA: double moving average; PVO percentage volume oscillator. For the OBV (on balance volume) linguistic variable the value obv_t for each day t is calculated from historical data using the algorithm: Initially at $t = 0$ $obv_0 = v_0$, then for each subsequent day t of historical data observations obv_t is calculated as follows: if $p_t > p_{t-20}$ then $obv_t = obv_{t-1} + v_t$; else if $p_t < p_{t-20}$ then $obv_t = obv_{t-1} - v_t$, else if if $p_t > p_{t-20}$ then $obv_t = obv_{t-1}$.

Name	Formula	Restrictions
SD 1		$\delta = 20$
SD 2	$sd(\ln(\frac{P_t}{P_{t-1}}), \ldots, \ln(\frac{P_{t-\delta}}{P_{t-\delta-1}}))$	$\delta = 50$
SD 3		$\delta = 100$
Bol 1	$Bol = \frac{p_t - ma_t}{2 \times sd(P_t, \ldots, P_{t-\delta})}$	$\delta = 20$
Bol 2		$\delta = 50$

Table 6.8: Breakout indicators. SD: standard deviation; Bol: Bollinger bands.

No.	Conditions
SMA buy	$(p_{t-\epsilon} < sma_{t-\epsilon})$ and $(p_t > sma_t)$
SMA sell	$(p_{t-\epsilon} > sma_{t-\epsilon})$ and $(p_t < sma_t)$
DMA buy	$(ma1_{t-\epsilon} < ma2_{t-\epsilon})$ and $(ma1_t > ma2_t)$
DMA sell	$(ma1_{t-\epsilon} > ma2_{t-\epsilon})$ and $(ma1_t < ma2_t)$
OBV buy	$(p_t > \max[p_{t-1}, p_{t-2} \dots p_{t-n}])$ and $(obv_t \leq \max[obv_{t-1}, \dots, obv_{t-n}])$
OBV sell	$(p_t < \min[p_{t-1}, p_{t-2} \dots p_{t-n}])$ and $(obv_t \geq \min[obv_{t-1}, \dots, obv_{t-n}])$
Vol DMA buy	$(vma1_{t-\epsilon} < vma2_{t-\epsilon})$ and $(vma1_t > vma2_t)$
Vol DMA sell	$(vma1_{t-\epsilon} > vma2_{t-\epsilon})$ and $(vma1_t < vma2_t)$

Table 6.9: Conditions for the linguistic variables to be defined. For the moving average variables a value ϵ refers to the period in days between the signal trigger (when the shorter average becomes lower or higher than the longer) and the day t for a signal to occur, it was fixed to 5 days in all tests.

base. Additional parameters specific to each linguistic variable given in table 6.9. These are all integer values. For mutation the value is incremented or decremented within a range of legal values (such that for example $ma_1 > ma_2$ etc) within a set interval of an absolute maximum and minimum that is from 20 to 260 days (i.e. 1 month to 1 year).

6.2.3 Membership Functions

In Chapter 4 the membership sets as well as the parameters to calculate the linguistic variables themselves are fixed and specified using initial parameters before optimization starts. In this section we provide a mechanism to adapt the number of membership functions used to describe the linguistic variable adaptively. The genotype is extended to contain a vector $noMF$ which has length equal to the number of possible linguistic variables. In other words the way the inputs are described is able to change. The extended genotype for the more adaptive rule base includes this vector to describe the membership functions and 4 integer values a, b, c, d which control properties of the actual calculation of the linguistic variables (see Figure 6.5).

To avoid excessive slowness in search processes a cache stores the values for the different of membership functions. The noMF vector acts as a key to the set of membership functions for a particular rule base. For example, if noMF $= 3, 7, \ldots, 5$ then the first lvar is described by 3 membership functions and the second by 7 and so on. A data cache is also used to store the linguistic variable values depending on the values of a, b, c, d as discussed in the previous section.

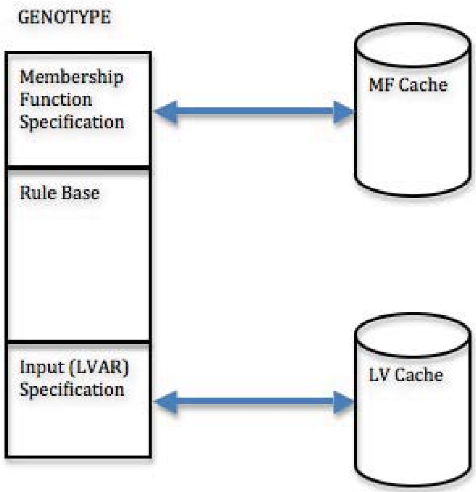


Figure 6.5: Genotype extended by additional parameters to control properties of the rule base. For optimization a cache is used to access each combination of membership function and linguistic variable only once.

6.3 Adapting the Solution

In contrast to the mechanisms in the previous section, this section introduces a separate aspect of adaptation more particular to the application domain. This is adapting the solutions produced by the system to the context of the portfolio being managed and the environment via consideration recent data and selection of training periods.

6.3.1 Search focus

Two repair operators are also used, see Table 6.3. The operator r-LEGAL is used to maintain solutions that specify a legal and defined chromosome rule base. The other repair operator, r-SAME, is used to maintain stability between generations (see Table 6.3). It is a binary operator with two rule bases as arguments and its effect is to modify the first genotype in such a way that it is no more than p percent different from a second genotype. On initialization the second genotype is the solution from the previous window and during the search it is the best solution found at the current generation.

The parameter p is adjusted depending on the performance of a real portfolio in relation to the index which serves as a benchmark. It is reduced when performance is worse than the benchmark and increased if the real portfolio is out performing the benchmark. The rationale is to focus the search close to solutions while they give good performance and to broaden the search when this performance decays. p is increased or decreased according to the formula:

$$p = \begin{cases} Sharpe \times 0.5 \times (1+k), & \text{if} \quad P_t^{real} < P_{t-d}^{real} \\ Sharpe \times 0.5 \times (1-k), & \text{if} \quad P_t^{real} > P_{t-d}^{real}, \\ 0, & \text{otherwise} \end{cases}$$

where the sharpe ratio $sharpe = \frac{(r_p - r_f)}{\sigma_p}$ and k is a constant set in the configuration file to control the sensitivity of p to variation in portfolio performance. The sharpe ratio combines measurement of both risk and return with respect to the benchmark to make the adjustment of p (controlling the level of restriction on the search) dependant not only on the difference between returns between the current and previous month, but also relative to benchmark returns that are expected to be achievable.

6.3.2 Training Data

The selection of the training data window controls the period of training data used when generating rule bases. Three methods for selecting a data window are able to be used by the system: static window; sliding window; variable length sliding window. The static window method uses a single initial period to evolve a rule base and then the rules from this period are used for all future trading. The sliding window uses a recent historical time windows for evaluation optimizing the rule base to recent periods. For variable length window the length of the window varies according to portfolio performance which relates benchmark returns to the return of a real portfolio found from applying the rule base and decreases the window length when performance is worse than the benchmark. Performance is measured in this case using the Sharpe ratio [98] of portfolio daily returns in the most recent trading period. The window length is calculated using the following formula:

$$windowlength = e^{k \times ln(sharpe)} \times maxLength,$$

where k is constant set to control the sensitivity of the window change which was set to 2, sharpe is a Sharpe ratio $sharpe = \frac{(r_p - r_f)}{\sigma_p}$ and $maxLength$ is the maximum window length. The new window date is never set earlier than the previous window start date. In the case that the the window length is calculated as earlier than this, the previous start date is used.

6.4 Impact of Adaptation on Performance

The adaptive methodologies discussed above are tested in managing a portfolio of stocks traded on the Australian Stock Exchange (hereafter ASX). Every month during the period of August 2001 to December 2006 two different portfolios are formed. The first one is created using a full adaptive evolutionary process, referred to as the "Adaptive EA" portfolio. The second one is created using a static rule base where the (single) rule base is generated for the first window and then used for the rest of the simulation. This is called the "Static EA" portfolio and serves as a comparison benchmark for the advantages provided by using an adaptive rule base. Finally, we also compare the performance of our EA portfolios to the ASX index, which reflects the performance of the market as a whole.

Figure 6.6 shows the portfolio values over the whole trading period, where the starting points have all been standardized to 1000. Both of the EA portfolios perform very well, clearly dominating the index. The total increase in portfolio value over the whole investment period is measured by "holding period return" reported in Table 6.10. The whole market during the investment period improves slightly with the Index's holding period return of only 123% over a period of more than 5 years. During the same time, the Static EA portfolio value increases by 310% and the Adaptive EA portfolio value increases by 474%. Adapting the system fully to the market conditions results in a higher value of the Adaptive EA portfolio than that of the Static EA portfolio by 1.5 times. Given the market conditions the annualized return, using either arithmetic or geometric calculation method, is very impressive for the EA portfolios. However, it should be noted that returns of the EA portfolios

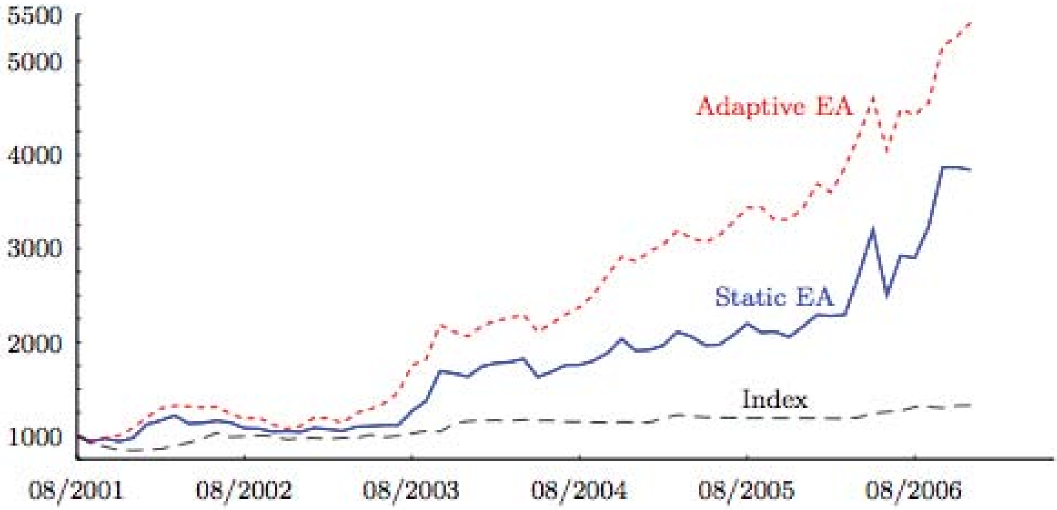


Figure 6.6: Value of the adaptive and static portfolios during the test period.

are more volatile than the market. Further analysis is required to confirm whether investors are rewarded with sufficient returns for the risk they bear.

Panel 2 of Table 6.10 reports statistics for excess portfolio values, ie. portfolio values above the risk free return. It can be argued, that over time, even if investors hold risk free assets, such as Treasury securities, they are rewarded with financial returns. Therefore, it is more relevant to investors to check the performance of the excess returns. Due to the increase in risk free returns over time, the excess returns of all portfolios are slightly lower than the raw returns. However, the performance relationship between different portfolios does not change.

The monthly returns of each portfolio are illustrated in Figure 6.7 while Table 6.11 further reports some characteristics of the return distributions. More than half of the time the Adaptive EA portfolio has a monthly return greater than 2.8%, whereas a half of the time the Static EA portfolio and the Index Portfolio have a return smaller than 2.4% and 0.8% respectively. Compared to the Index portfolio, the EA portfolios occasionally do experience higher loss. The largest negative returns for the two EA portfolios are 21.8% and 11.9%, whereas the largest loss for an index portfolio is only 8%. The index portfolio does not experience any loss greater than 10% in any given month, compared to the occurrence frequency of 3.15% and 1.56%

Table 6.10: Portfolio Returns.

Panel 1. Raw portfolio value

	ASX Index	Static EA	Adaptive EA
Holding Period Returns	123.34%	310.06%	473.99%
Annualized Arithmetic Returns	14.31%	28.59%	34.06%
Annualized Geometric Returns	13.67%	25.48%	32.08%
Annualized Volatility	11.39%	25.51%	20.45%

Panel 2. Excess portfolio value

	ASX Index	Static EA	Adaptive EA
Holding Period Returns	117.99%	304.7%	468.63%
Annualized Arithmetic Returns	8.96 %	23.25%	28.71 %
Annualized Geometric Returns	8.32 %	20.13%	26.72 %
Annualized Volatility	11.36 %	25.49%	20.45 %

for the two EA portfolios. This result may be indicative of the penalty function not sufficiently discarding the choice of stocks that are more likely to experience a large decline. However, both of the EA portfolios exhibit positive skewness, which is what investors prefer. A positive skewness portfolio has a high probability of large returns. It should be noted that the EA portfolios also have lower kurtosis, which implies less regular and smaller swing away from the mean in investor returns. Overall, the Adaptive EA portfolio has better return potential compared to the Static EA portfolio while maintaining a lower level of risk.

We have noted that our EA portfolios have much better return potential than the Index, and at the same time are more volatile. The Sharpe ratio [98] measures how much excess returns (portfolio return r_p above the risk free rate r_f) investors are awarded for each unit of volatility, ie.

$$\text{Sharpe} = \frac{r_p - r_f}{\sigma_p}.$$

As can be seen in Table 6.12, the Sharpe ratio for the Index is only 0.79, whereas that for the Static EA portfolio is 0.91. Adaptive EA portfolio has the best Sharpe

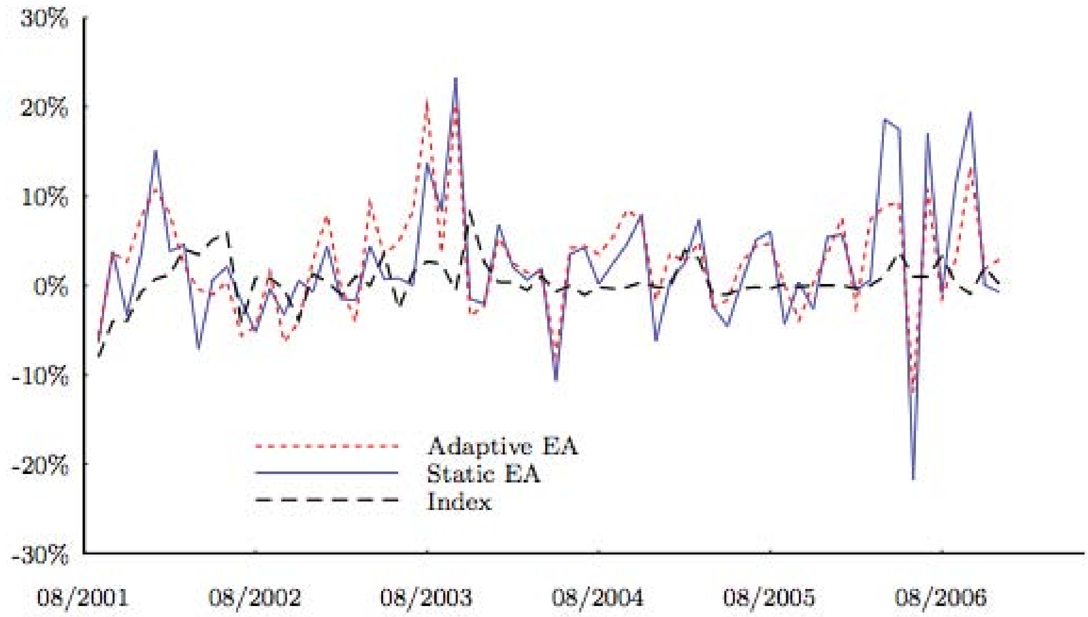


Figure 6.7: Monthly returns of the adaptive and static portfolios during the test period.

ratio of 1.4, nearly double the reward investors receive for holding a passive index portfolio. The improvement in return potential for a fully adaptive system has been well above some additional level of risk for investors.

The Sharpe ratio focuses on portfolio volatility which measures total risk of the portfolio. Modern portfolio theory further decomposes volatility into systematic risk and unsystematic risk. The systematic risk component reflects how the changes in market conditions affect portfolio values, whereas the unsystematic risk component is unique to each portfolio. The constraints under which we form portfolios, such as the maximum stocks in each country or each sector, are in fact constraints to build well diversified portfolios. A well diversified portfolio should have return awarded to compensate for the systematic risk component only. Denote $r_{m,t}$ the returns at time t of the market, the systematic risk β_p of portfolio p is determined by the Capital Asset Pricing Model (CAPM) equation:

$$r_{p,t} - r_{f,t} = \alpha_p + \beta_p(r_{m,t} - r_{f,t}) + e_{i,t}. \tag{6.1}$$

Since the excess return $r_{p,t} - r_{f,t}$ of any portfolio should be fully explained by its level

Table 6.11: Portfolio return characteristics.

	ASX Index	Static EA	Adaptive EA
Average monthly return	0.470%	2.383%	2.838%
Median monthly return	0.078%	0.685%	2.791%
Largest positive return	8.071%	23.257%	20.367%
Largest negative return	-8.008%	-21.769%	-11.930%
Skewness	-0.0829	0.3759	0.4356
Kurtosis	2.7062	2.1657	1.2516
Frequency of gain greater than 10%	0.000%	12.500%	7.813%
Frequency of loss greater than 10%	0.000%	3.125%	1.563%

of systematic risk β_p and the market risk premium $r_{m,t} - r_{f,t}$, in an efficient market the alpha value of the portfolio, α_p, should be zero. If it is not and in fact there is a positive value then the portfolio is outperforming relative to its level of systematic risk and the performance of benchmark index. The higher the alpha value, the better the portfolio is to hold. Both of the EA portfolios have positive alpha, indicating a superior performance (see Table 6.12). The Adaptive EA portfolio has a value of alpha 1.5 times larger than that of the Static EA portfolio.

Table 6.12: Standard portfolio performance measures.

	ASX Index	Static EA	Adaptive EA
Sharpe ratio	0.789	0.912	1.404
Alpha	NA	0.124	0.184
Information Performance Rank	NA	0.572	1.160
Selectivity	NA	0.124	0.184
Net Selectivity	NA	0.031	0.126

The robustness of alpha values can also be measured through the information performance rank. Essentially, it evaluates the active stock-picking skills of the strategy, once unsystematic risk generated from the investment process is accounted for. The formula and an extended interpretation for the of this measure is provided in Chapter 5.

The information ratio of the Static EA portfolio is 0.57, indicating a very good performance. The Adaptive EA portfolio has an information rank of 1.16, double that of the Static EA portfolio. The information rank very close to 1 is indicative of a very strong and consistent performance.

Both EA portfolios have a positive Selectivity, the Adaptive EA portfolio has much stronger performance than the Static EA one. After accounting for the difference in total risk using the Net Selectivity measure, the Static EA portfolio still has a positive Net Selectivity, but is quite marginal. On the other hand, the Adaptive EA portfolio has a substantial positive Net Selectivity of 0.13, again indicating a very strong performance. The details of the Selectivity and Net Selectivity measures are provided in Chapter 5.

To track variation in the performance of the portfolios over time, rolling Alphas and rolling Information Ratios are used. Each month new Alpha and Information Ratio are calculated based on the data for the previous year. Figure 6.8 and 6.9 graph these two series. Even though both of the EA portfolios have positive overall Alphas and larger than 0.5 Information Ranks as shown in Table 6.12, only the Adaptive EA portfolio is able to maintain a consistent level of strong performance over time. This clearly shows the advantages of adapting the portfolio to changing market conditions.

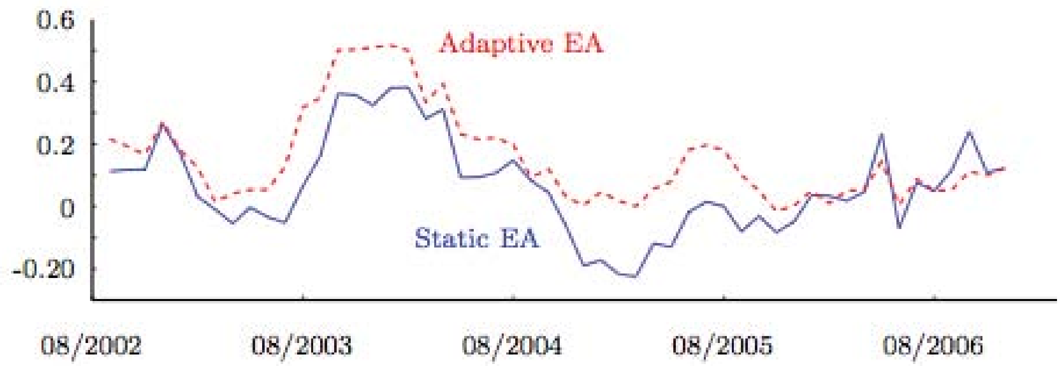


Figure 6.8: Change in alpha of the adaptive and static portfolios during the test period.

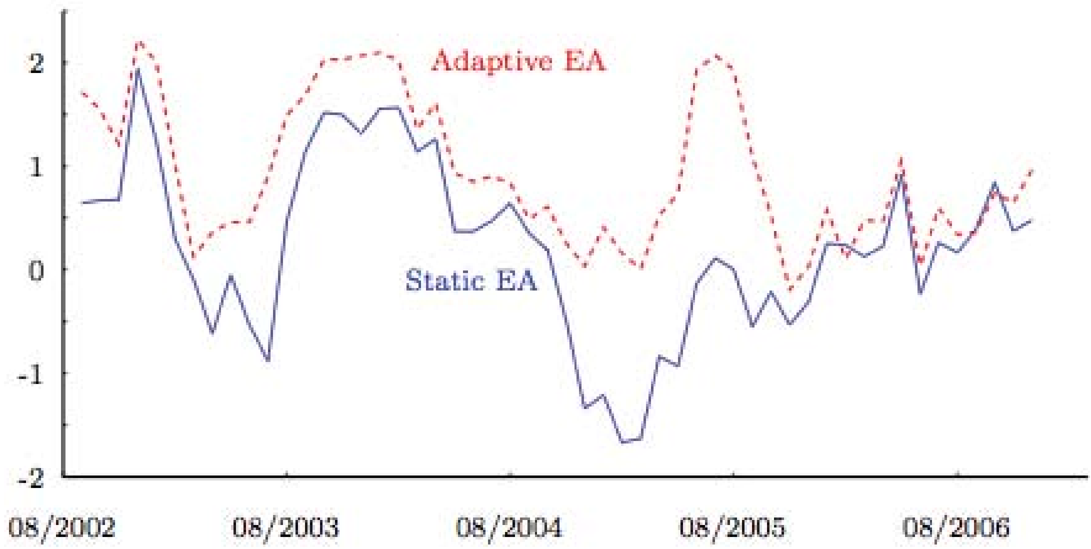


Figure 6.9: Change in the information ratio of the adaptive and static portfolios during the test period.

Chapter 7

Optimization and Prediction

This chapter provides an analytical approach to fuzzy rule base optimization. While the rest of this thesis and indeed most research in the area has been done experimentally, theoretical considerations give new insights to the task. Using the symmetry that is inherent in the fuzzy rule base formulation method used, we show that the problem of finding an optimal rule base can be reduced to solving a set of quadratic equations that generically have a one dimensional solution space. This alternate problem specification can enable new approaches for rule base optimization. We examine one possible approach resulting from the analysis.

7.1 Overview of the Problem

A fuzzy rule is a causal statement that has an *if-then* format. The *if* part is a series of conjunctions describing properties of some linguistic variables using fuzzy sets that, if observed, give rise to the *then* part. The *then* part is a value that reflects the consequence given the case that the *if* part occurs in full. A rule base consists of several such rules and is able to be evaluated using fuzzy operators to obtain a value given the (possibly partial) fulfilment of each rule.

Membership functions are a crucial part of the definition as they define the mappings to assign meaning to input data. They map crisp input observations of linguistic variables to degrees of membership in some fuzzy sets to describe properties of the linguistic variables. Suitable membership functions are designed depending on the specific characteristics of the linguistic variables as well as peculiar properties related to their use in optimization systems. Triangular membership functions (see

Figure 7.2) are widely used primarily for the reasons described in [91]. Other common mappings include ‘gaussian’ [55] and ‘trapezoidal’ [39] membership functions. The functions are either predefined or determined in part or completely during an optimization process. A number of different techniques have been used for this task including statistical methods, heuristic approaches [6], and genetic and evolutionary algorithms [42, 82, 29, 99]. Adjusting membership functions during optimization is discussed in [42, 107].

In this work fuzzy rule bases are optimized in an evolutionary process to find rules for selecting stocks to trade. A rule base that could be produced using this system could look as follows:

- If *Price to Earnings Ratio* is *Extremely Low* then $rating = 0.9$
- If *Price Change* is *High* and *Double Moving Average Sell* is *Very High* then $rating = 0.4$

The *if* part in this case specifies some financial accounting measures (Price to Earnings ratio) and technical indicators [2] used by financial analysts; the output of the rule base is a combined rating that allows stocks to be compared relative to each other. In that system rule bases were evaluated in the evolutionary process using a function based on a trading simulation.

The task of constructing rule base solutions that is considered includes determining rule statements, membership functions (including the number of distinct membership sets and their specific forms) and possible outputs. These parameters and the specification of data structures for computational representation have a significant impact on the characteristics and performance of the optimization process. Previous research in applications [96, 25, 39] has largely consisted and relied upon experimental analysis and intutition for designs and parameter settings. This section takes a theoretical approach to the analysis of a specific design of a fuzzy rule base optimization system that has been used in a range of successful applications [80, 55]; we utilize the symetries that are inherent in the formulation to gain insight into the optimization. This leads to an interesting alternate viewpoint of the problem that may in turn lead to new approaches.

In particular, our formal definition and framework for the fuzzy rule base turns the optimization problem into a smooth problem that can be analyzed analytically. This analysis reduces the problem to a system of quadratic equations whose solution space has the surprising property that it generically contains a whole line. It should be possible to utilize this fact in the construction of fast and efficient solvers, which will be an important application of this research. The approach in this paper builds on experimental research presented earlier in the thesis. But it should be noted that a number of other mechanisms have been proposed for encoding fuzzy rules [39].

The methods we consider could also be used in an evaluation process where the error is minimized with respect to fitting rule bases to some training data — in the context of the above example this would allow a system to learn rules with an output that is directly calculated from the data. For example a rule base evaluated in this way could be used to forecast the probability that a stock has positive price movement in some future time period. A rule in such a rule base could look like: If *Price to Earnings Ratio* is *Extremely Low* and *Double Moving Average Buy* is *Very High* then *probability of positive price movement* is 0.75.

7.2 Rule Base Solution Representation

This section provides details of the processing and interpretation of inputs after modeling using linguistic variables. A rule base representation has both a literal fuzzy logic representation and an internal genotype representation (used in an evolutionary algorithm). To compare rule bases an evaluation function is required.

7.2.1 Literal representation

The core of the prediction model comprises sets of fuzzy rules that encode information about linguistic descriptions of the input factors. A fuzzy rule is a propositional statement with a formal *If — then* structure. The *if* part of each rule specifies a relationship between the linguistic descriptions. And the *then* part is a weighting given the complete satisfaction of a rule.

First order knowledge is not captured directly in the representation that is considered here. However for practical implementation, quantifier restrictions are implied by the scope of the rulebases application and the learning process by specifying the range over which predicates are able to be applied. For example a rulebase is able to learned for a particular time, market, or, possibly, an industry sector or other type of share.

Let us introduce some precise definitions of what is meant by the *rule base solution representation*. First of all, we are given L linguistic variables $\{A^1, ..., A^L\}$. Each linguistic variable A^i has M_i linguistic descriptions $\{A^i_1, ..., A^i_{M_i}\}$ that are represented by triangular membership functions μ^i_j, $j = 1, ..., M_i$. A *fuzzy rule* has the form

$$\text{If } A^{i_1} \text{ is } A^{i_1}_{j_1} \text{ and } A^{i_2} \text{ is } A^{i_2}_{j_2} \text{ and } \cdots \text{ and } A^{i_k} \text{ is } A^{i_k}_{j_k} \text{ then } o, \tag{7.1}$$

where $i_1, ...i_k \in \{1, ..., L\}$, $j_k \in \{1, ..., M_{i_k}\}$ and $o \in [0, 1]$.

A *rule base* is a set of several rules. Let us assume that we are given a rule base consisting of n rules:

$$\begin{array}{lll}
\text{If } A^{i_1^1} \text{ is } A^{i_1^1}_{j_1^1} \text{ and } A^{i_2^1} \text{ is } A^{i_2^1}_{j_2^1} \text{ and } \cdots \text{ and } A^{i_{k_1}^1} \text{ is } A^{i_{k_1}^1}_{j_{k_1}^1} & \text{then} & o^1 \\
\text{If } A^{i_1^2} \text{ is } A^{i_1^2}_{j_1^2} \text{ and } A^{i_2^2} \text{ is } A^{i_2^2}_{j_2^2} \text{ and } \cdots \text{ and } A^{i_{k_2}^2} \text{ is } A^{i_{k_2}^2}_{j_{k_2}^2} & \text{then} & o^2 \\
\vdots & & \vdots \\
\text{If } A^{i_1^n} \text{ is } A^{i_1^n}_{j_1^n} \text{ and } A^{i_2^n} \text{ is } A^{i_2^n}_{j_2^n} \text{ and } \cdots \text{ and } A^{i_{k_n}^n} \text{ is } A^{i_{k_n}^n}_{j_{k_n}^n} & \text{then} & o^n,
\end{array}$$

where $i_l^m \in \{1, ..., L\}$ and $j_l^m \in \{1, ..., M_{i_l^m}\}$. Given a vector $x \in \mathbb{R}^L$ of observed values, whose components are values for the linguistic variables $A^1, ..., A^L$, we can evaluate the rule base as follows: the function ρ describes the way the rule base interprets data observations x to produce a single output value. This value has an application specific meaning and can be taken to be a real number (usually normalized to lie between zero and one). More precisely, ρ is defined as follows:

$$\begin{array}{rcl}
\rho : \mathbb{R}^L & \to & \mathbb{R} \\
x = \begin{pmatrix} x^1 \\ x^2 \\ \vdots \\ x^L \end{pmatrix} & \mapsto & \dfrac{\sum_{m=1}^{n} o^m \prod_{l=1}^{k_m} \mu^{i_l^m}_{j_l^m}(x^{i_l^m})}{\sum_{m=1}^{n} o^m}.
\end{array}$$

For example if a rule base produced by the system is specified as follows:

- If *Price Oscillator* is *High* then *rating* $= 0.9$
- If *Price to Book Value is Low* is *High* and *Alpha* is *High* then *rating* $= 0.4$

The resulting mapping is applied to a set of assets:

$$asset_1, asset_2, \ldots, asset_i.$$

The result is an output rating for each one that is a prediction of percentage return over a subsequent period (such as one week, a month, etc). Suppose that data observations are read for Price Oscillator = 0.6, Price to Book Value = 0.7 and Alpha=0.01. Initially output levels are converted to fuzzy values, for instance if membership functions for Price Oscillator are as illustrated in Figure 7.1 below such that data point 0.6 maps to a level of 0.7 High. In the case that the other

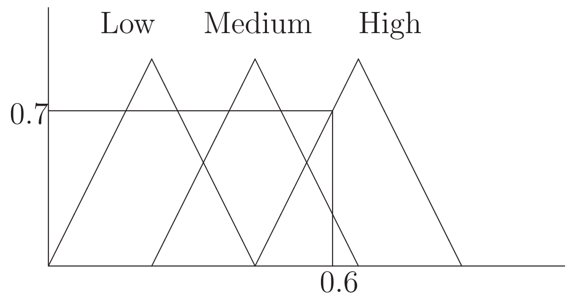


Figure 7.1: Membership function mapping from data observation 0.6 to membership degree 0.7.

observations are similarly processed to obtain 0.5 degree High for Price to book value and Alpha 0.0 High, then the rule base output rating is:

$$((0.9 * 0.7) + ((0.7 * 0.5 + 0.5 * 0) * 0.4))/(0.7 + 0.4)$$

$$= (0.63 + 0.14)(1.1) = 0.7.$$

The next section describes the evolutionary procedure implemented to learn rule models of this form.

7.2.2 Genotype representation

For the optimization process rule bases are encoded as candidate solutions using 2 matrices in which rows correspond to individual rules and columns to particular linguistic variables.

The first matrix encodes the input parts of a rule base with a maximum number of rules n, and up to L linguistic variables by a matrix of positive integers, $I = I(L; n)$. The i, j-th element indicates a membership function μ applicable for the j-th linguistic variable to be used in the i-th rule, if the integer is 0 then it is interpreted to mean that variable is not used. The output parts (for a rule base with n rules) are encoded using a vector O containing all possible values for o, where the i-th element is the output (or weight) of the i-th rule; values for o are discrete levels from the set $D = \{0, 1/d, \ldots, 1\}$. A 0 value indicates that the weight for the rule is 0, i.e. it is not used in the rule base. The number of individual slots or genes in this representation is therefore the total number of elements in I and O is $n(L+1)$. More precisely we have

$$I_{m,r} = \begin{cases} j_s^m & \text{if } i_s^m = r \text{ for some } s \\ 0 & \text{else} \end{cases} .$$

For example, the first row of I would be of the form

$$0 \ \cdots \ 0 \ j_1^1 \ 0 \ \cdots \ 0 \ j_2^1 \ 0 \ \cdots \ 0 \ j_{k_1}^1 \ 0 \ \cdots \ 0,$$

where j_1^1 is the i_1^1-th entry, j_2^1 is the i_2^1-the entry and so forth. The vector O has the form

$$O = \begin{pmatrix} o^1 \\ o^2 \\ \vdots \\ o^n \end{pmatrix} .$$

7.3 Evaluation Function

We consider an evaluation function (to minimize) that measures the error when training a rule base to fit a given data set. This is slightly different from the simulation or ranking measures used elsewhere in the thesis, however it is an abstraction for analysis that enables meaningful results and also is useful in its own right. This *training data* consists of a set $\{x_i, y_i\}_{i=1\ldots N}$, where each

$$x_i = \begin{pmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^L \end{pmatrix}$$

is a vector that has as many components as there are linguistic variables, i.e. $x_i \in \mathbb{R}^L \ \forall \ i = 1, ..., N$, and each y_i is a real number, i.e. $y_i \in \mathbb{R} \ \forall \ i = 1, ..., N$. Then the evaluation function has the form

$$\begin{aligned}
\epsilon &= \sum_{i=1}^{N} (\rho(x_i) - y_i)^2 & (7.2) \\
&= \sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{n} a_{ij} o^j}{\sum_{j=1}^{n} o^j} - y_i \right)^2, & (7.3)
\end{aligned}$$

where

$$a_{sm} = \prod_{l=1}^{k_m} \mu^{i_l^m}_{j_l^m}(x_s^{i_l^m}).$$

Our aim is to optimize the rules base in such a way that the evaluation function ϵ becomes minimal. This involves two separate problems. Firstly, the form of the membership functions μ^i_j may be varied to obtain a better result. Secondly, the rule base may be varied by choosing different rules or by varying the weights o^i. In this analysis we will concentrate on the second problem, taking the form of the membership functions to be fixed. For example, we can standardize the number of membership functions for each linguistic variable A^i to be $M_i = 2n_i - 1$ and define

$$\mu^i_j = \begin{cases} 0 & : \ x \leq \frac{j-1}{2n_i} \\ 2n_i x + 1 - j & : \ x \in \left[\frac{j-1}{2n_i}, \frac{j}{2n_i}\right] \\ -2n_i x + 1 + j & : \ x \in \left[\frac{j}{2n_i}, \frac{j+1}{2n_i}\right] \\ 0 & : \ x \geq \frac{j+1}{2n_i} \end{cases}$$

for $j = 1, ..., 2n_i - 1 = M_i$. These functions are shown in Figure 7.2.

Moreover, we can consider the number n of rules to be fixed by either working with a specific number of rules that we want to consider, or by taking n to be the number of all possible rules (this number will be enormous, but each rule whose optimal weight is zero, or sufficiently close to zero can just be ignored and most weights will be of that form), depending on the application. The resulting optimization problem will be considered in 7.4.2.

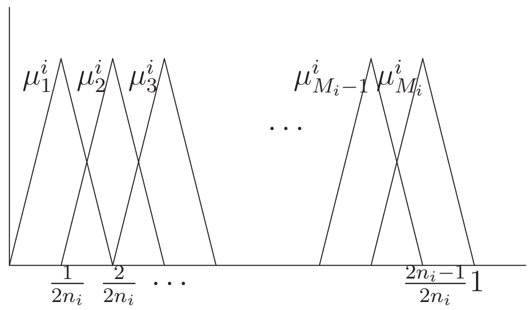


Figure 7.2: Membership Functions

7.4 Analysis

This section contains the detailed analysis of the problem described in Section 7, we gratefully acknowledge the coauthors of the paper [59] for mathematical assistance with the remainder of this chapter. Firstly, we determine the maximum possible number of rules and then consider the optimization problem for the evaluation function. As a result, we are able to reduce the optimization problem to a system of equations (7.9), that has the remarkable property that it allows (generically) a one-dimensional solution space. This is the content of Theorem 7.4.1.

7.4.1 Search space

The search space is the set of all potential rule base solutions. Let us first of all compute the maximum number of rules $n_{\max}$ that we can have. Each rule can be written in the form

$$\text{If } A^1 \text{ is } A^1_{j_1} \text{ and } A^2 \text{ is } A^2_{j_2} \text{ and } \cdots \text{ and } A^L \text{ is } A^L_{j_L} \text{ then } o,$$

where in this case $j_i \in \{0, 1, ..., M_i\}$ and $j_i = 0$ implies that the linguistic variable A^i does not appear in the rule. Then we have

$$n_{\max} = (M_1 + 1) \times (M_2 + 1) \times \cdots \times (M_L + 1) - 1.$$

Note that we have subtracted 1 to exclude the empty rule. If we include the possible choices of weights o^i with discretization $o^i \in \{0, \frac{1}{d}, ..., 1\}$, then we have a system of

$$(d + 1)^{n_{\max}}$$

possible rule bases.

7.4.2 Optimization problem

In this subsection we will treat the optimization problem described in 7.3. We have to take the training data $\{x_i, y_i\}_{i=1\dots N}$ and the various membership functions μ_j^i as given, so we can treat the various a_{ij} as constants and simplify

$$
\begin{aligned}
\epsilon(o) &= \sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{n} a_{ij} o^j}{\sum_{j=1}^{n} o^j} - y_i \right)^2 \\
&= \sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{n} (a_{ij} - y_i)^2 o^j o^j + 2 \sum_{j<k} (a_{ij} - y_i)(a_{ik} - y_i) o^j o^k}{\left(\sum_{j=1}^{n} o^j\right)^2} \right) \\
&= \frac{\sum_{j=1}^{n} A_{jj} o^j o^j + 2 \sum_{j<k} A_{jk} o^j o^k}{\left(\sum_{j=1}^{n} o^j\right)^2} \\
&\quad \text{with } A_{jk} = \sum_{i=1}^{N} (a_{ij} - y_i)(a_{ik} - y_i) \\
&= \frac{\sum_{j=1}^{n} \sum_{k=1}^{n} A_{jk} o^j o^k}{\left(\sum_{j=1}^{n} o^j\right)^2}.
\end{aligned}
$$

We want to find weights o^i such that this expression becomes minimal. In our formulation this requirement is smooth in the o^i, so we can compute the partial derivatives of the evaluation function with respect to the weights. At a minimal point $o_{\min} \in \mathbb{R}^n$, we must have

$$
\frac{\partial \epsilon}{\partial o^1}(o_{\min}) = 0, \frac{\partial \epsilon}{\partial o^2}(o_{\min}) = 0, ..., \frac{\partial \epsilon}{\partial o^n}(o_{\min}) = 0.
$$

It will turn out that this requirement is equivalent to a system of quadratic equations. So let us compute

$$
\begin{aligned}
\frac{\partial \epsilon}{\partial o^q}(o) &= 2 \frac{\left(\sum_{i=1}^{n} A_{iq} o^i\right) \left(\sum_{k=1}^{n} o^k\right) - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} o^i o^j}{\left(\sum_{i=1}^{n} o^i\right)^3} \qquad (7.4) \\
&= \frac{2}{\left(\sum_{i=1}^{n} o^i\right)^3} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} (A_{iq} - A_{ij}) o^i o^j \right). \qquad (7.5)
\end{aligned}
$$

If we can simultaneously solve these n equations

$$\frac{\partial \epsilon}{\partial o^1}(o) = 0, \frac{\partial \epsilon}{\partial o^2}(o) = 0, ..., \frac{\partial \epsilon}{\partial o^n}(o) = 0,$$

then we have found a local extrema. For only two rules, for example, we obtain

$$\begin{aligned}
\frac{\partial \epsilon}{\partial o^1}(o) &= \frac{2o^2}{(o^1+o^2)^3}\left((A_{11}-A_{12})o^1 + (A_{21}-A_{22})o^2\right) \\
\frac{\partial \epsilon}{\partial o^2}(o) &= \frac{2o^1}{(o^1+o^2)^3}\left((A_{12}-A_{11})o^1 + (A_{22}-A_{21})o^2\right)
\end{aligned}$$

Therefore, if we assume that $o^1 \neq 0$ or $o^2 \neq 0$, then the optimal solution is

$$o^1 = \frac{A_{22}-A_{21}}{A_{11}-A_{12}}o^2.$$

This is a whole line that intersects zero in $\mathbb{R}^2$. This phenomena can be seen clearly in the following picture:

More than two rules

If we have more than two rules, then the conditions become

$$\frac{\partial \epsilon}{\partial o^q} = 0 \Leftrightarrow \left(\sum_{i=1}^{n}\sum_{j=1}^{n}(A_{iq}-A_{ij})o^i o^j\right) = 0,\ q = 1, ..., n. \tag{7.6}$$

Theorem 7.4.1. *Generically, there exists a one-parameter family of solutions to the system (7.9). Hence the space of extremal points for ϵ is a line in $\mathbb{R}^n$ that passes through zero.*

Proof. We will show that the n equations (7.9) are dependent, i.e. that we only need to solve $n-1$ of these equations and the n-th equation then follows automatically. For this purpose, we rewrite the system

$$\begin{aligned}
\left(\sum_{i=1}^{n}\sum_{j=1}^{n}(A_{iq}-A_{ij})o^i o^j\right) &= \sum_{\substack{j=1 \\ j\neq q}}^{n} o^j \underbrace{\left((A_{qq}-A_{qj})o^q + (A_{jq}-A_{jj})o^j\right)}_{B_{qj}} \\
&+ \sum_{\substack{j=1 \\ j\neq q}}^{n} \sum_{\substack{i=1 \\ i\notin\{q,j\}}}^{n} \left((A_{iq}-A_{ij})o^i o^j\right).
\end{aligned}$$

Note that

$$B_{qj} = -B_{jq}.$$

Denote the q-th equation by E_q. Using the equality above, we compute

$$\begin{aligned}\sum_{k=1}^{n} o^k E_k &= \sum_{k=1}^{n} \sum_{\substack{j=1 \\ j \neq q}}^{n} \underbrace{B_{kj} o^k o^j}_{=0} \\ &\quad + \sum_{k=1}^{n} \sum_{\substack{j=1 \\ j \neq q}}^{n} \sum_{\substack{i=1 \\ i \notin \{q,j\}}}^{n} \left(\underbrace{(A_{ik} - A_{ij})}_{C_{ijk}} o^i o^j o^k \right) \\ &= 0.\end{aligned}$$

The last term vanishes due to the fact that the tensor C_{ijk} is symmetric in the index pair (i, j), symmetric in the index pair (i, k) and skew (i.e. anti-symmetric) in the index pair (j, k). Such a tensor has to vanish identically. It is hence sufficient to solve (7.9) just for $(n-1)$ equations, the last equation is automatically satisfied. $\square$

We have successfully reduced the problem of finding optimal weights o^i for a rule base (given an arbitrary set of training data points) to a system of n equations for n unknowns, where n is the number of rules. Moreover, we have shown that the space of extremal points for the evaluation function is a line through the origin in $\mathbb{R}^n$. Hence a genetic algorithm will be able to find an optimal solution in $[0, 1]^n$ using well-established and fast methods such as may be found in [25]. The reason for this, somewhat surprising, result lies in the specific form of our rule base formulation: the values of the weights themselves are not important, but the relationship that they have with respect to each other is. Mathematically, the optimal solution o is really an element of $(n-1)$-dimensional projective space $\mathbb{RP}^{n-1}$, rather that an element of $\mathbb{R}^n$.

7.5 Simple Implementation

In this section we discuss a possible application of this analysis for reducing the size of the search space. We would like to minimize the error in predicting asset price movements using a rule base. Training data comprises price series for the assets and associated observations of linguistic variables.

An evaluation function measures error between observed percentage price movement in training data and prediction by rules. The rule base output is interpreted as a prediction of percentage price movement. Note that rule base output originally in the interval $[0.0, 1.0]$, is scaled by a factor δ appropriate for the application. Scaling it to an interval $[0.0, 0.2]$ would have the meaning that the highest output should predict price movement of $0.2 \Rightarrow 20\%$ rather than $1.0 \Rightarrow 100\%$. The training data consists of a set $\{\vec{x_i}, y_i\}_{i=1\ldots N}$, where each

$$\vec{x_i} = \begin{pmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^L \end{pmatrix}$$

is a vector constructed from the factors given in 8.1 such that each $x_i^{1\ldots L=30}$ is an observation of $f_{1\ldots L=30}$ (it is a pre-condition that $x_i \in \mathbb{R}^L$), and each y_i is a real percentage price change ($y_i \in \mathbb{R}$). The evaluation function is a measure of squared error between predicted (ρ) and actual (y_i percent return:

$$\begin{aligned} \epsilon &= \sum_{i=1}^{N} (\delta\rho(x_i) - y_i)^2 &(7.7)\\ &= \sum_{i=1}^{N} \left(\delta \frac{\sum_{j=1}^{n} a_{ij} o^j}{\sum_{j=1}^{n} o^j} - y_i \right)^2, &(7.8) \end{aligned}$$

where

$$a_{sm} = \prod_{l=1}^{k_m} \mu_{j_l^{im}}^{i_l^{im}}(x_s^{i_l^{im}}).$$

We optimize the rules to minimize ϵ. The number of training points N is a result of the length of the . The procedure for a rule base fitness evaluation given a training window of historic data with length, len and a time interval, s, over which predictions are fixed is as follows:

Algorithm 1 Evaluation

$\epsilon \leftarrow 0$
for all (assets a_i in a market) **do**
  for $(t \leftarrow s; t < len; t \leftarrow t + s)$ **do**
    $\epsilon \leftarrow \epsilon + (\delta\rho(x_{a_i}) - p_{a_i,t}/p_{a_i,t-s})^2$
  end for
end for

where $p_{a_i,t}$ is the price of an asset at time t. The number of training data points, N, is equal to the number of assets times the window length divided by the forecast horizon or step s. Figure 7.4 shows some of the price series for stocks listed in the ASX200 that are considered simultaneously in the application during historic data for the four years from 2004 until 2008. In the experimental analysis discussed in this paper a forecast window of $s = 5$ days (one working week) was considered.

The optimization relies on a population based methodology in which individuals (rule bases) are evolved in a process emulating natural selection. Genotypes are represented using three arrays I, U and O. I is an $m \times n$ matrix of integers where each $i - jth$ element corresponds to a membership function μ in $i - th$ rule for the $j - th$ linguistic variable, each variable has the same number of membership function specifications so for five membership functions the possible values of I are $1, 2, 3, 4$ or 5. U is an $m \times n$ matrix of Boolean values, if $U_{i,j} = TRUE$ then the input in $i - th$ rule for the $j - th$ variable is switched on and used in the linguistic rule description, otherwise it is not used. O is a vector of double values with size m, one for each rule, and corresponds to the output levels, a value of 0 means the rule has zero weight and is not used.

Mutation and crossover operators are applied to vary the genotype. They are defined as follows: mutate inputs — select a random gene from either I or U with uniform probability and (with equal probability) either replace with a new random value or in/de-crement by 1 step with equal probability; mutate outputs — select and replace an output at random; crossover — uniform crossover over I and U; and rule crossover — swap rules between two different individuals, at the genotype this means the whole $i - th$ from I, U, O are swapped.

The main rule base optimization procedure acts on the genotypes and comprises steps as follows:

Algorithm 2 EA

Require: $P_0, r_{best}, gen, operators[], opProb[], ooProb$
  while $(gen <$ max_gen$)$ **do**
    parents[] $\leftarrow$ selectParents(P_{gen}) {t size=2}
    operator $\leftarrow$ $selectOperator(operators, opProb)$
    offspring $\leftarrow$ $applyOperator(parents)$
    if $(random() < ooProb)$ **then**
      $adjustOutput$(offspring)
    end if
    P_{gen} $\leftarrow$ replaceWorse(P_{gen}, offspring) {t size=2}
    $r_{cbest} \leftarrow best(P_{gen})$
    opProb[]$\leftarrow$update(opProb[])
    ooProb$\leftarrow$update(ooProb)
    if $(r_{cbest} > r_{best})$ **then**
      $r_{best} \leftarrow r_{cbest}$
    end if
    $gen \leftarrow gen + 1$
  end while
  return r_{best}

The operator probabilities are updated dependent on the success (obtaining a better solution) from the operators during the run as discussed in Chapter 6. The probability of using the separate output optimization procedure oscillates in between *never* and *always* being applied $[0, 1]$ a set number of times (5 in the experiments here). Fitness is assigned in the method applyOperators() using algorithm 1. In this way rule base antecedents (*if* parts of each rule) and consequents (*then* parts) separately for optimization. At every step it is possible that a separate optimization of rule base outputs occurs with probability $ooProb$.

The separation is accomplished using the theorem from the previous section which contains the details and proof. Recall that we are able to rewrite the evaluation function as follows:

$$\begin{aligned}\epsilon(o) &= \sum_{i=1}^{N}\left(\frac{\sum_{j=1}^{n} a_{ij}o^j}{\sum_{j=1}^{n} o^j} - y_i\right)^2 \\ &= \frac{\sum_{j=1}^{n}\sum_{k=1}^{n} A_{jk}o^j o^k}{\left(\sum_{j=1}^{n} o^j\right)^2},\end{aligned}$$

where $A_{jk} = \sum_{i=1}^{N}(a_{ij} - y_i)(a_{ik} - y_i)$. The alternate objective for the output part of the rules (the vector O in the genotype) is found by taking derivatives of the this expression to restate the optimization problem in terms of the output for a particular input specification. This system is as follows:

$$\frac{\partial \epsilon}{\partial o^q} = 0 \Leftrightarrow \left(\sum_{i=1}^{n} \sum_{j=1}^{n} (A_{iq} - A_{ij}) o^i o^j \right) = 0, \; q = 1, ..., n. \tag{7.9}$$

The evaluation function for the outputs of the rule base is now to minimize this system. We solve the problem for the outputs by minimizing the expression. Recall that n is the number of rules, then in the implementation, the constants $A_{j,k}$ are calculated as follows (note that for simplicity of expression the loops for accessing the training data set by asset and day are omitted and we refer to the implied training data set consisting of pairs $(\vec{x_i}, y_i)$:

for $i = 1 \dots N$ **do**
  $tmp \leftarrow 0$
  for j = 1. . . n **do**
    for k=1 . . . n **do**
      . . .
      $tmp = tmp + (a_{i,j} - y_i)(a_{i,k} - y_i) \dots$
      . . .
    end for
  end for
  $A[j][k] = tmp$
end for

Using $A[][]$ rule bases are evaluated by minimizing 7.9. Note that the constants only need to be calculated once for each output optimization because the output weight optimization occurs while the input is constant.

7.6 Experimentation

This section contains experimental results obtained by application of the procedures rule learning procedures described in the previous section. Table 7.6 lists the evolutionary algorithm settings that were used. In the implementation the output was optimized separately with oscillating probability set by $P = cur\_gen/max\_gen \bmod 3$. In addition the rule base was repaired to maintain < 4 inputs per rule and < 5

active rules (i.e. having output greater than zero). 5 discrete output weights were possible in the specification. Test data was sourced from Data Stream International (http://www.datastream.com/) and consisted of ASX200 listed stocks and associated data for the period 2006 to 2008 inclusive.

A steady state algorithm with elitism was used in the normal case and altered by the addition of a local output optimization subroutine (leaving other aspects such as adaptive operator probabilities found to increase fitness intact). The fitness objective was set to minimize the squared error between predicted asset price movement (from the output of fuzzy rules) and the real movement over one week periods. Two penalties were in effect – one to penalize rules that do nothing (always predict zero change); and a second, that balances the other penalty, to penalize rules that incorrectly anticipate change direction.

Parameter	Value
P. OO Adj.	0.0 – 1.0 / 3 times
Population	500
Generations	7500 — 10000
Elitism ?	Yes
Selection	Tournament $size = 2$
Horizon	5 days
Win len.	120 days
Initial Operator P.	0.3333 (3 Operators)
Penalties	No ranking = 0.05
	Direction = 0.1
Max Rules	5
Max Inputs	4
Output levels (d)	5

Table 7.1: System parameters.

The experiments are designed to provide a fair comparison of system performance with local output optimization and the usual case and examine the differences that are induced in the application when the local output optimization procedure is used. Specifically we examine the best fitness values, the runtime in main EA cycles and changes in prediction performance. Hypothetically, it seems plausible that separate output optimization will cause the following to occur:

- The EA could converge with fewer cycles of full fitness evaluation since the

	Norm.	OO adj.
Min.	-341.1837	-242.3193
Max.	-149.2366	-100.4559
Median	-220.766	-161.1760
Mean.	-226.8101	-162.7535
Sdev	56.55486	45.69378
Mann Whit.	–	W =205
		$p < 0.01$ (gt.)

Table 7.2: Paired comparison (with different data) for 7,500 generations.

	Norm.	OO adj.
Min.	-281.3545	-239.5838
Max.	-62.44117	-100.4559
Median	-160.6017	-155.1053
Mean.	-170.1535	-158.8335
Sdev	63.10455	45.00803
Mann Whit.	–	W = 125
		p = 0.2375 (gt.)

Table 7.3: Paired comparison (with different data) for 10,000 generations.

outputs are no longer a part of the genotype that is evolved but is set in effect deterministically instead.

- The search may proceed faster also because of improvement in the ability to select input genotypes. I.e. the potential of particular linguistic variables could be assessed earlier in the search. This is because individuals that use inputs which lead to higher fitness when the output is set to the optimal level would be favored for reproduction over those with inputs that are not able to be improved by setting the outputs.

- Better parts of the search space could be more fully explored by the directing the population to contain better solutions (i.e. that are more fit by the measure used in evaluation). In the normal case it is possible that information is likely to be lost by random variation.

Testing of the EA was conducted using longer and shorter termination conditions using different training data. Tables 7.2 and 7.3 summarize the best fitness distributions after 7,500 generations and 10,000 generations for 30 runs. Figures 7.5 and 7.6 show box plots of these distributions. Table 7.4 shows comparison of actual error from applying best solutions outside training data.

Non-parametric statistical testing is used to compare the results in a sound way without assumptions about the distribution of the series. The Mann-Whitney test counts the number of pairs from two series (s_{1_i}, s_{2_i}) where the s_{1_i} is greater than or less than s_{2_i} [47], or in the two-sided test greater *or* less than. The significance is obtained by examination of the p statistic, it is standard that a p level less than 0.05 is interpreted as sufficiently significant to reject the null hypothesis that the series are not different. All testing was with paired, each pair comparing the new method and the normal method for a specific data window.

If the termination condition is to stop after 7,500 generations the improvement is, empirically, very clear. The local output optimization (OO) test produced solutions with better fitness than the normal case (N) Table 7.2 with a p-level less than 0.1 percent. The median and mean fitness for the OO runs were also over 35% higher. In addition, the distribution for the OO showed a lower range and and standard deviation indicating that it produces more consistent results. Specifically, the range was 192 for the OO compared with 142. Figure 7.5 summarizes these comparisons in a box plot, in particular highlighting the stability and higher fitness from different runs obtained using the OO.

If the search is run over 10,000 generations the improvement is less significant (see Table 7.3). The mean for the OO was 9% better, however the median was almost the same. By the Mann-Whitney test there is insufficient evidence to conclude there is a real difference between the results. It is still the case, however, that the OO best results exhibited less variance and more consistency. Figure 7.6 provides a visual comparison of the best fitness distributions resulting from runs of the OO and N cases.

Table 7.4 provides analysis comparing (absolute) prediction error that was found on applying the rules outside the training data. The results provided were from applying the rules for 5000 different predictions of 201 assets considered in the investment universe. For each optimization run, the best rule base was applied to predicting the price change of each stock three times over the subsequent 15 day period after the end of the training window; for each application the difference between the actual and the predicted change was recorded. As would be expected from the fitness results, it was the case that generalization performance of solutions

	Norm.	OO adj.
$7,500$ generations		
Mean.	0.0406	0.03810
Sdev	0.0382	0.03335
Mann Whit.	–	W = 13256668
		$p < 0.01$ (2 sided)
$10,000$ generations		
Mean.	0.0405	0.0409
Sdev	0.03653	0.03563
Mann Whit.	–	W = 12278316
		p = 0.1246 (2 sided)

Table 7.4: Error outside training data.

produced by the OO method were better in the $7,500$ generation test runs. The mean error was lower (3.8% vs. 4%) and the improvement was significant at the 5% probability threshold (by the p value).

However, for the tests at $10,000$ generations it was found the results for generalization were comparable. And also it was the case that the increase in training fitness did not translate to better predictions, this is possibly due to over fitting the model to test data. But, surprisingly, it was found that the best prediction performance was for the OO method in the shorter run. This indicates that by using the new method the search process was altered in a way to more quickly converge to rule bases with higher generalization ability in shorter runs, but that the advantage (in generalization) is lost as higher fitness solutions with poorer generalization characteristics come to dominate the population.

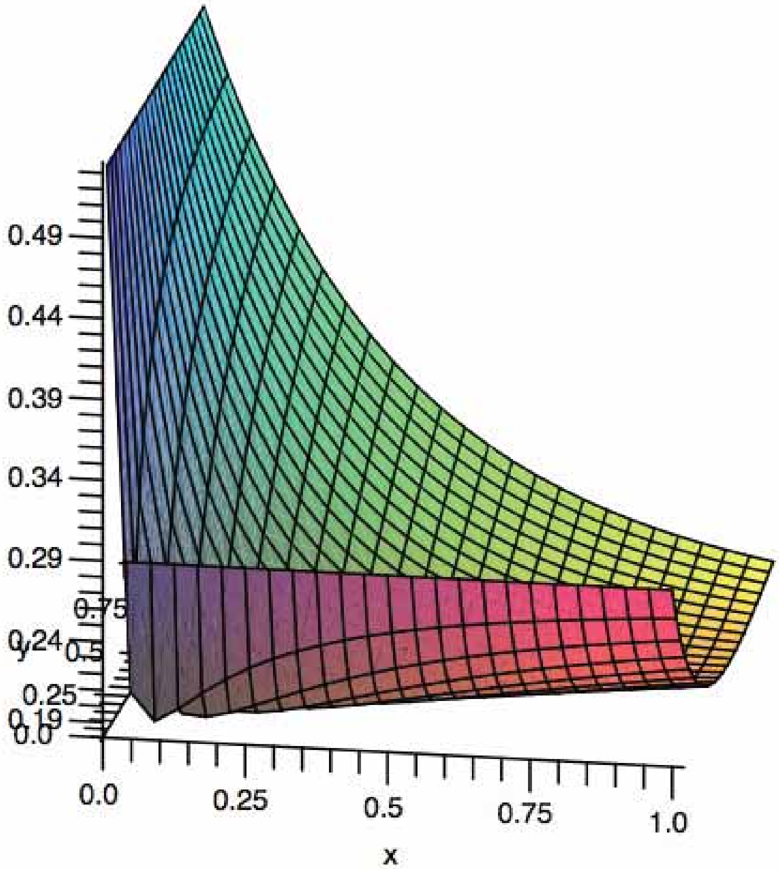


Figure 7.3: Evaluation function for two rules

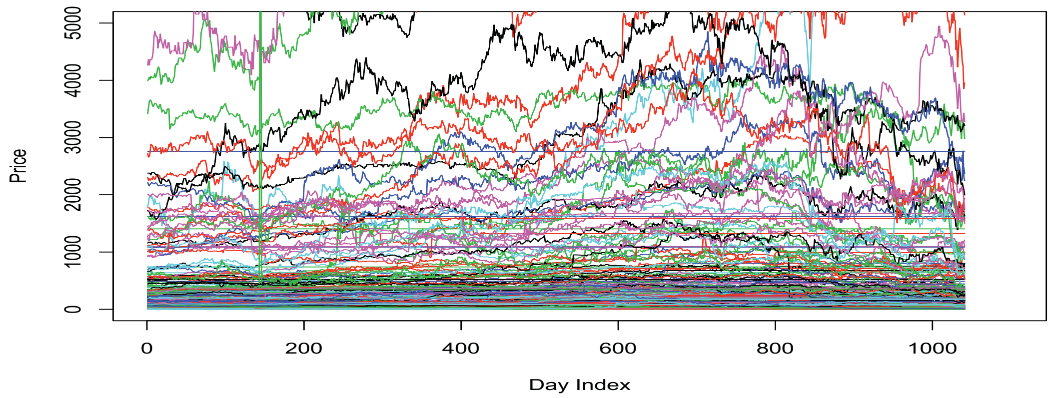


Figure 7.4: A selection of the asset price time series that are the object of the prediction.

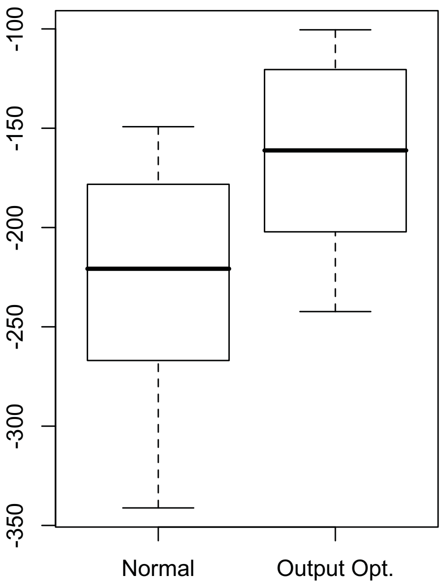


Figure 7.5: Box plot of the sample fitness values for the test 1 (7500 generations) showing the first.

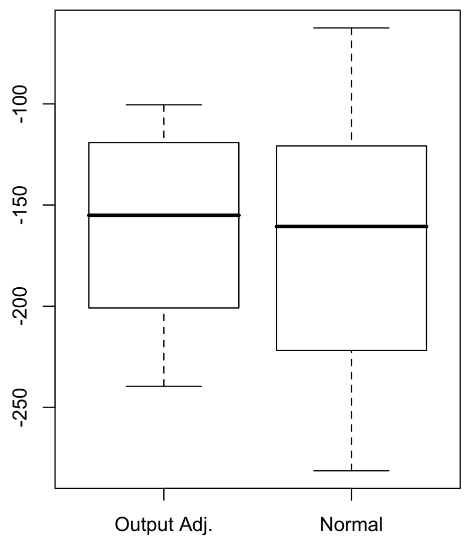


Figure 7.6: Box plot of the sample fitness values for test 2 (10,000 generations).

Chapter 8

Fundamental Analysis, Multiple Objectives

This Chapter describes two significant extensions of the approach developed in Chapters 4 – 6. There are relatively few studies which make use of fundamental data sources in intelligent model construction. In this chapter we add selected additional variables to the input data set. These model inputs are designed to facilitate a financial "reasoning" process. Considerations include sector analysis and change over time of variables. The second major extension discussed is the incorporation of multiple objectives. This is achieved through aggregate rule base solutions which are optimized to different objectives. The components of an aggregate solution recommendation are found from separate optimization routines that differ in evaluation function and training data set. In relation to these additions we provide some mechanisms to handle issues related to the large input data set.

8.1 Analysis Style

It is a common procedure in finance to compare stocks by a various criteria, either in isolation or simultaneously. As discussed in Chapter 2, the most common ways utilize linear regression of the CAPM model described in Chapter 2 – an *alpha* strategy – or by methods such as discussed in Section 8.2 below. Our method enables a *non linear* prediction model which relates asset performance to a large set of explanatory variables. Several rule bases are aggregated to build recommendations that contain consideration of different forecast horizons and both risk and return criteria. Using the same approach it is also possible to construct "hierarchal" rule base structures where different model inputs are used in different elements of an aggregate model.

The model inputs or factors are one of the kernels of the methodology, particularly from a financial view point. This is because the relationships between these factors and returns are essentially the possibilities implied in the model definition. Another fundamental aspect of the definition in a Computational Intelligence model is the set of parameters that specify the optimization process. We assume and implement the principles for equity markets and in the discussion we use the term stock. However various derivative instruments such as options, warrants or indeed any market listed item, are able to be traded using the recommendations produced by the intelligent decision support system. In this section we discuss the definitions of the $L = 30$ model factors, $f_i, \ldots, f_L$, that are used in the implementation.

We incorporate three distinct analysis styles:

- Portfolio theory,
- Technical analysis,
- Fundamental analysis.

Portfolio theory involves optimizing portfolios using the theoretic asset pricing models. The second two approaches involve constructing factor valuation models. For the procedure of technical analysis price and volume data derived from the market itself is used in a specialized form of time series analysis. Fundamental analysis is a more broad technique in the information considered and comprises study of the firms underlying listed stocks, the operating environments implied by different industry sectors, macro economic conditions and so forth. Files containing raw observations of of basic data for each stock during an historic data period are processed to produce model factors, Table8.1. The basic input types are listed in Chapter 2. Let us briefly discuss the incorporation of the different types of model factor in the remainder of this section.

The system considers the capital asset pricing model for securities in factors 1 and 2 (Table 8.1). It does not directly optimize these values by selecting stocks to produce an optimal portfolio for a given risk profile, but rather uses this information to attempt to identify under priced stocks — as stocks with a high alpha are underpriced by the market according the pricing model. Specifically, the alpha and beta for a stock are calculated as follows. First denote $r_{m,t}$ the returns at time t of the market, and $r_{s,t}$ is stock return], the systematic risk β_p of stock p is determined by the regression model:

$$r_{s,t} - r_{f,t} = \alpha + \beta(r_{m,t} - r_{f,t}) + e. \tag{8.1}$$

Fundamental valuation of underlying firms whose stock is listed in the market involves the Factors 3 to 24 in Table8.1. Several company attributes are derived from the input data files and processed to produce model factors. In particular input for each stock is adjusted to obtain a value relative to other stocks in the same sector by calculating for each input element $X_{s,t}$ with a value for a stock s on day t a value related to the other stocks in the same sector denoted by $X_{1,t}, \dots, X_{k,t}$: $X\text{-sector}_{s,t} = X_{s,t}/(X_{1,t} + \dots + X_{k,t})$. Furthermore, another important attribute of the series is the change over time. This characteristic is captured by the model input factors from the sector adjusted value using a calculation: $X\text{-sector-change}_{s,t} = X\text{-industry}_{s,t}/(X\text{-sector}_{s,t-period})$. The normalization within industry sectors is enforced because stocks within each sectors have common attributes shared with other businesses in the same operating environment. the growth or decline with respect to the sector and in general is tracked by the momentum factors. In addition a variable industry sector identifies a stocks sector and also the relative placement in the sector with respect to market capitalization are included here.

The Factors 25 to 30 are technical indicators used by financial analysts for valuation in practice. They are designed to extract characteristics of price and volume time series for prediction on the basis of extracting an interpretation of participant behavior when interacting with the market, in effect these processes involve applying different types of filters to the price and volume series. The percentage price oscillator and price volume oscillator emphasize cyclical patterns. The oscillators are calculated by taking the ratio between a longer and shorter moving averages. Standard deviation is a running standard deviation with length three months. Longer and shorter period price change factors track the rate of change in price (price momentum). Bollinger bands are lines one standard deviation from the mean around a stocks price – commonly sell or buy signals are specified if the price moves outside the upper and lower deviations. The money flow index (MFI) attempts to track the rate of capital flow into and out of stocks by relating price and volume. The calculation is with respect to an arbitrary period variable p a values at time t are calculated using the following steps:

1. If $price_{t-p} > price_{t-p-1}$ then $MF_t^+ = MF_{t-1}^+ - (price \times volume)$,

2. If $price_{t-p} < price_{t-p-1}$ then $MF_t^- = MF_{t-1}^- - (price \times volume)$,

3. If $price_t > price_{t-1}$ then $MF_t^+ = MF_{t-1}^+ + (price \times volume)$,

4. If $price_t t < price_{t-1}$ then $MF_t^- = MF_{t-1}^- + (price \times volume)$,

5. $MFI_t = MF_t^+ / MF_t^-$.

8.2 The Systematic Representation of Known Strategies

This section provides details of a number of well known stock strategies that use the extended factor set provided in Table8.1. Again the basic approach to strategy implementation is on selection of assets in an investment universe of potential investments (for example, a market or industry sector). Comparison entails ranking potential choices by desirability in some manner according to a criteria.

The procedure of "stock screening" is a method to refine the investment universe by criteria that define desirable or favorable assets to hold [70]. Two types of screen are discussed: *sequential* and *simultaneous*. In sequential screening criteria are progressively applied one after the other to eliminate stocks that do not fit the specification. Simultaneous screening involves applying all criteria at once to rank stocks by a score that is a combined measure such as with a weighted average. Stock screening is relatively easy to implement and is the basis of a number of stock services for investors, for example figures 8.1 and 8.2 show web based services in which users are able to specify, based on absolute values, criteria to narrow an investment universe.

Using stock screens well known strategies are able to be implemented and applied systematically. Let us discuss some examples.

Warren Buffet is the chairman of Berkshire Hathaway and also a famous investment guru. His approach concentrates on selecting and holding companies below their intrinsic value which may be defined using criteria including P/E ratio relative to comparable firms, high book to market value, free cash flow to equity and measures of growth potential [44]. A stock screen to implement this may have the following criteria [70].

1. Market capitalization is in the top 30% of the universe,

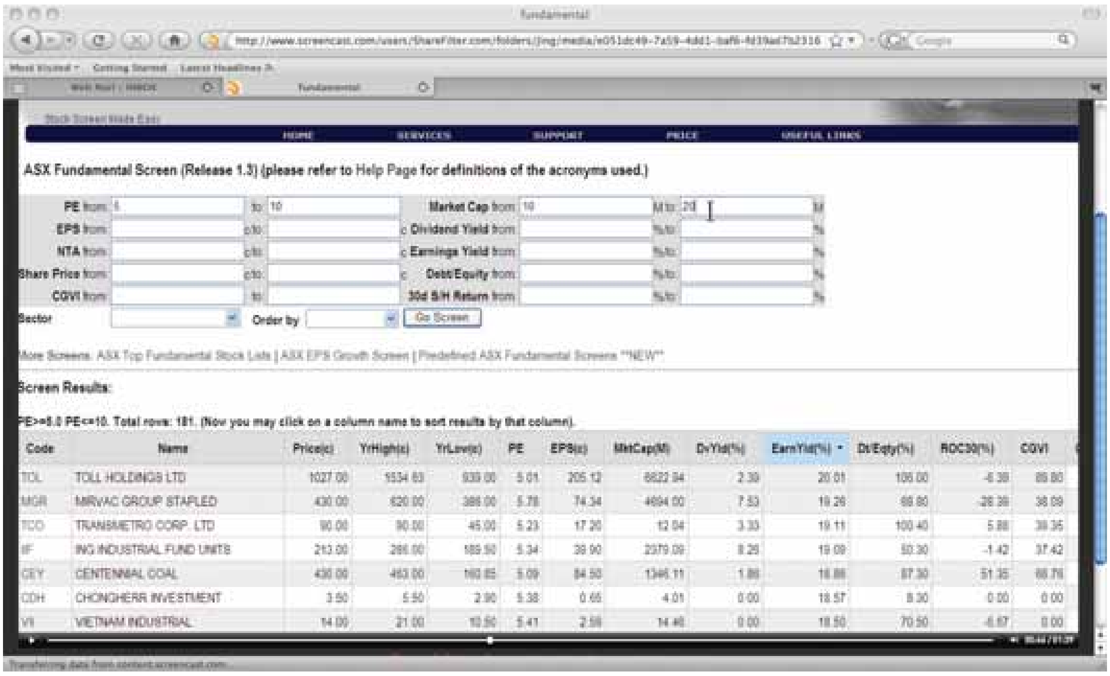


Figure 8.1: The subscription services provided by Share Filter (www.sharefilter.com) to filter and screen stocks in the ASX involves sequential stock screening.

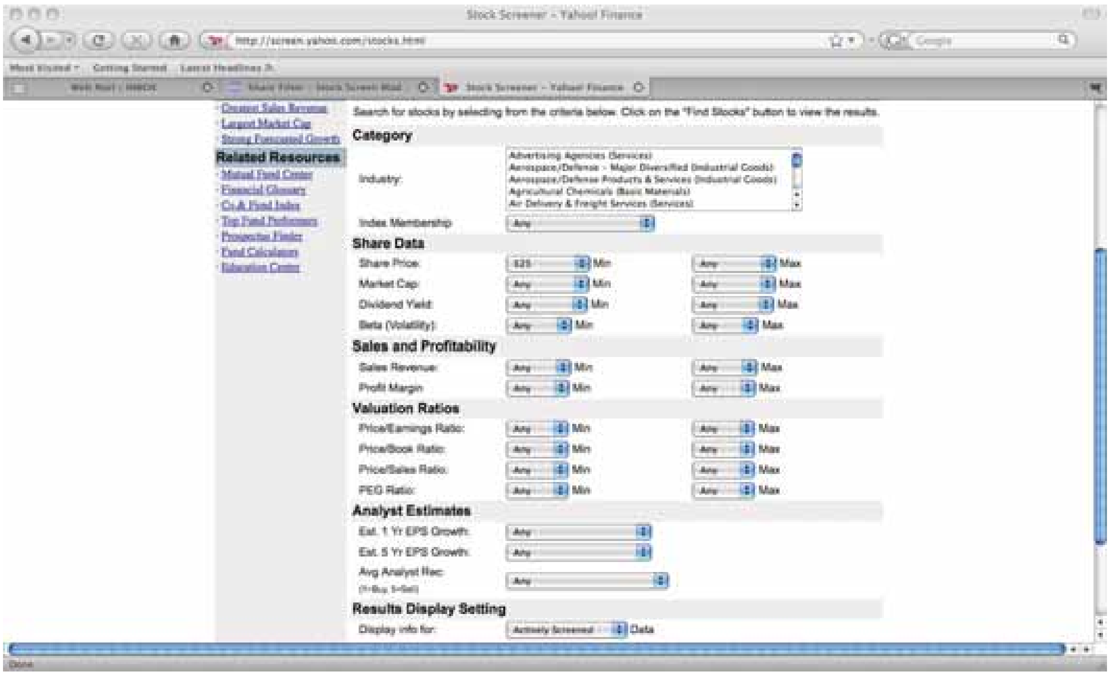


Figure 8.2: More recently Yahoo (screen.yahoo.com/stocks.html) has provided a similar though less customizable web based service.

2. Return on equity is greater than 15% over the previous 3 years,
3. Free cash flow is in the top 30% of the universe,
4. Growth rate in the market value is greater than the growth rate in the book value.

Josef Lakonishok wrote an influential paper [61] contributing the "contrarian investment" strategy which is based on the belief that companies go in and out of favor with investors and good value companies will return to favor in time. To measure whether companies are good value, in the sense of contrarian investment, it is possible to compare information from ratios including price to book value, price earnings ratio, and earnings per share, to industry averages (to ensure comparisons make sense and reflect considerations unique to equity classes). However, rather than just selecting constantly poor performing companies it is necessary to identify companies that are starting to rebound using momentum information. A contrarian investment strategy is able to be implemented using the following screen [70]:

1. Market capitalization is in the top 50%,
2. PE ratio is less than the median for the industry,
3. Forecast EPS for next year is greater than forecast was for the current year,
4. EPS forecast has been revised upwards in the last month,
5. Return is greater than the return of the index for last 6 months.

These strategies are able to deal with a large investment universe in a systematic fashion. More importantly, the strategies in [44] and [61] are both expressed in natural language in some detail, however they are able to transformed into a set of criteria able to applied for selection of potential investments using rules. The same can be done for numerous other strategies so long as information is available. We notice that in most cases two specific types of information are used regarding the basic information: change/growth over time and relative value to other assets in an industry sector or the market. In a few cases absolute values are used (market capitalization). This is reflected in the choice of inputs for the rule base strategy we implement.

It is possibly desirable to apply the criteria simultaneously rather than sequentially because otherwise potentially good choices may be eliminated early. A common method to do this is by constructing an "aggregate Z - score" [41]. To calculate a z-score for factor k for a stock i:

$$z_{i,k} = \frac{X_{i,k} - \mu_k}{\sigma_k},$$

where $X_{i,k}$ is the value of the factor k for the stock i, μ_k is the mean of all observations of the factor k and σ_k is the standard deviation. The aggregate z-score for a stock i and a number of factors $k = 1, 2, \ldots, K$ is then:

$$Z_i = 1/k \times (z_{i,1} + \ldots z_{i,K})\,.$$

All these methods are able to be precisely represented into fuzzy descriptions using appropriate membership sets. The factors given in Table8.1 calculate measures of these from the raw data sources. Chapter 6 discusses adaptive mechanisms to change the time windows and other variables used in parameter calculation. The specifications of the top or bottom $x\%$ of the market and so on are able to follow naturally from the method used to calculate membership sets using the procedure given in chapter 4 and developed to be adaptive in chapter 6. For example if there are 10 membership functions defined then each set contains 10% of observations. By using different membership specifications, the meaning can be refined to better reflect the intentions of the original strategies expressed in natural language. Finally, we reiterate that by aggregating solutions learned from different training data horizons and for different evaluation functions we extend the capacity for representing and learning strategies computationally.

8.3 Multiple Investment Objectives

To construct recommendations multiple fuzzy rule base systems are optimized by a hybrid heuristic consisting of an evolutionary algorithm and deterministic local search procedures. Information from the resulting solutions is then collated into recommendations for decisions relating to assets that incorporate multiple objectives. A combined recommendation consists of two lists of assets from the universe of possible choices: one containing assets that are recommended to buy and the other to sell. These lists are input to a kind of decoder which determines transactions that should be implemented for portfolio management.

The literature provides examples of research into aggregating fuzzy rule bases for classification and control applications by approaches such as voting and fuzzy set operations. These approaches have been successfully applied in a variety of cases, for instance see [108, 97]. In this paper the focus is on an evolving fuzzy system for financial prediction involving numerous asset price series. The literature also describes approaches assisted by computation for asset valuation and selection. Such approaches include: multiple linear regression for factor modeling [33]; asset filtering [70]; and intelligent heuristic approaches (for instance see [37, 36, 14]). In general regression modeling, filtering and other quantitative approaches developed by financial experts, see [70], have a number of limitations such as linearity, inability to readily adapt and requiring detailed participation of an expert user and process designer.

In a hybrid heuristic learning approach integrating an evolutionary algorithm and local search techniques we develop an adaptive asset valuation model with elements of many financial analysis methods. The input factors provide an ability for the system to approximate these approaches. The method involves a model that automatically update itself in terms of the factors considered and relationships among them [37].

8.4 Aggregate Solution Evaluation

A rating for a stock by a rule base on a day t is obtained using observations of the values of the model factors for the stock on the day t. Let us use a subscript to denote this rating such that $rating_{s,t}$, means the rating for a particular stock s on day t. A ranking is defined here a set of stocks ordered by some value associated with each stock. In this case let us assign this value to be the output of a rule base given data for the stock on a particular day. For a set of stocks $M = \{s_1, s_2, \ldots, s_m\}$ a rule base can be applied to order a set of stocks to obtain a ranking of all stocks in the set on the day t as follows:

$$R_t^\rho(M) = [(s_{1,t}, \rho(\vec{x}_{s_1,t}), \ldots, (s_{m,t}, \rho(\vec{x}_{s_m,t})] \,, \tag{8.2}$$

where $\rho(\vec{x}_{s_i,t}) \geq \rho(\vec{x}_{s_{i+1},t})$. Each element of a ranking is a pair comprising the ranked stock and its rating, $(s_{i,t}, \rho(\vec{x}_{s_i,t}))$, has rank $i \in \mathbb{Z}, i \geq 1$.

The fitness of a rule base is defined to be a measure of its ability to rank stocks by return ordering over a specific period of time in the future termed a forecast horizon of length H days. An ideal return ordering $R_{ideal,t}$ for a day t is constructed by looking forward H days into the future within the available training data. Then we find the average price for a stock, $p_{s,H}$ during a P day period starting after the H'th day in the training data window. As this operation takes place in training data we may assume that for every stock the return during the period is simply:

$$r_{s,t,H} = p_{s,H} - p_{s,t},$$

where $p_{s,H}$ is the average price over a period starting H days after t. The reason the average price is used is to avoid the fitness being overly sensitive to fluctuations in stock price on particular dates. An ideal ranking for days in the training window for comparing the rule base ranking with is able to be defined as follows:

$$R_t^{ideal}(M) = \left[(s_{1,t}, r_{s_1,t,H}), \ldots, (s_{m,t}, r_{s_m,t,H})\right]. \tag{8.3}$$

Incorporating this calculation evaluation functions are defined to compare rankings $R_{\rho,t}$ from rule bases with $R_{ideal,t}$. In the application rule bases are tested using input data from several days so that rule base fitness is not overly dependent on patterns in a single day of training data.

As a step to constructing an evaluation function let us initially define a comparison operator for comparing rankings. An obvious method for comparing the ordering of A and B is to count the number of times the same stock has an identical rank in both. However it is preferable that the method should be more lax for a number of reasons, including for accuracy in that two rankings would be defined as very different if the rankings were out of sync by even a single element; trying to find rules to predict a very specific ranking property would likely lead to over fitting and loss of generalization; and in addition the ordering of stocks within the top percentile is not relevant since all stocks in this group are, relative to the others, recommended to buy. For these reasons and to make the optimization task easier, we use a flexible approach for comparison designed to be sensitive to very small changes in similarity due to any change in the ranking order.

Given two rankings A and B that order stocks in a set $M = \{s_1, s_2, \ldots, s_m\}$ we define two corresponding sub rankings:

$$a = [(s_{a_1}, r_{s_{a_1}}), \ldots, (s_{a_{u_1}}, r_{s_{a_{u_1}}})],$$

and

$$b = [(s_{b_1}, r_{s_{b_1}}), \ldots, (s_{b_{u_2}}, r_{s_{b_{u_2}}})],$$

with sizes $u_1, u_2 < m$ containing the highest u_1 and u_2 rated stocks in A and B respectively. We construct two sets of stocks which are subsets of M

$$a_s = \left\{s_{a_1}, \ldots, s_{a_{u_1}}\right\}$$

and

$$b_s = \left\{ s_{b_1}, \ldots, s_{b_{u_2}} \right\}.$$

A real value measure of similarity of two rankings A_M, B_M defined over the set of m stocks M is then found by the operator:

$$\text{similarity} : R_M \times R_M \mapsto \mathbb{R}$$

$$\text{similarity}_{u_1,u_2}(A_M, B_M) = \frac{\mid a_s \cap b_s \mid}{\max(|a_s|, |b_s|)}. \tag{8.4}$$

where $u_1, u_2 \in \{u \in \mathbb{Z} | 0 \leq u \leq m\}$. The meaning is interpreted as the number of stocks from the top u_1 of A_M that are also in the top u_2 of B_M.

Now we define the evaluation function that uses a training window of length $horizon + 2 \times period$ where H is the forecast horizon and P is a fixed period of sequential days that is both the number of days used to test the rule base and also the number of days used to calculate average values for the ideal ranking. Let the first day in the training window be denoted day T then:

$$eval_{buy,H}(\rho_B) = \sum_{t=0}^{P} \frac{\text{similarity}_{l,q}(R^{\rho_B}_{T+t}, R^{ideal}_{T+P})}{P}, \tag{8.5}$$

where, $R^{\rho_B}_t$ is a ranking from a rule base ρ_B with respect to a set of listed stocks and R^{ideal}_{T+P} is an ideal ranking of stocks at a day taken at the end of the possible training testing days. The parameters l and q may be tuned by experimentation or adaptively. In optimization it is easier to try to find l top stocks using the rule base that are in the q of the ideal ranking if $l > q$. Another fitness function is also used in the system to measure the ability of rule bases to rank stocks by likelihood of decreasing value. This function is defined in a similar way, the only difference is that the order of the ideal ranking is reversed:

$$eval_{sell,H}(\rho_B) = \sum_{t=0}^{P} \frac{\text{similarity}_{l,q}(R^{\rho_B}_{T+t}, \text{reverse}(R^{ideal}_{T+P}))}{P}. \tag{8.6}$$

Figure 8.3 shows a visualization of the fitness of an evolved rule base according to this fitness evaluation method. The green squares show stock selections in the rule base ranking on test days which are correctly in the top percentage of the ideal ranking. Red squares are those in the lowest 50% of the ideal ranking, and yellow are neither in the bottom two percentiles of of the ideal nor in the top as is desired.

Figure 8.3: Visualization of the phenotype fitness from the rankings implied using a rule base solution in test data. The vertical axis shows the test day in the training window and the horizontal axis the ranking performance of stocks with respect to a target ranking.

8.5 Search Process for a Single Solution

The number of possible rule bases able to expressed using the *If-then* grammar of fuzzy rules is very large. In general, for a single rule with q inputs (it is not necessary that all L factors are active in a particular rule hence the terms i and n) with d possible output ratings is:

$$p = d \sum_{i=1}^{n} q^i \begin{pmatrix} q \\ i \end{pmatrix}.$$

Therefore, the total number of rule bases containing m rules that can be written is of the order p^m. In this case we have 30 inputs, and 10 possible outputs and up to 20 rules which gives search space size significantly greater than 10^{100}.

To handle this large search space the rule bases are optimized in stages during which the solution is fixed and then extended in steps (see Figure 8.4). In the first stage the best single rule with two inputs is found by exhaustive search. This best rule is fixed (set to be a compulsory first rule) for all members of the population for an evolutionary search process, see [37]. An initial limited genotype with a small number of rules is then optimized by the EA. The best from this search is extracted and the genotype is extended by one rule while the previous best is fixed as a component for the whole population. This procedure of adding rules is repeated until the rules that are produced start to become less general or the maximum number is reached (the measure used for generality was simply the number of stocks ranked, i.e. don't result in a zero output rating).

Initially we evaluate the C_2^L single rules that are possible combinations of two factors. A percentage of the best of these are inserted into the initial population before the evolutionary search process as single rules with the maximum output value, this rule may also be fixed across the population for the whole search by action of an operator $r-$FIXED which alters a genotype such that a certain percentage are (uniformly) switched be the same as a prototype individual.

The algorithm consists of the following steps:

1. Initialize population $P = \langle \rho_1, \rho_2, \ldots, \rho_n \rangle$ of n rule base individuals where $g\%$ are from the enumeration search and the remainder are random

2. Initialize variables from parameter file: *cBEST*, *BEST*, *SWI*, *generation*, *FIXED*, $rules_{max}$

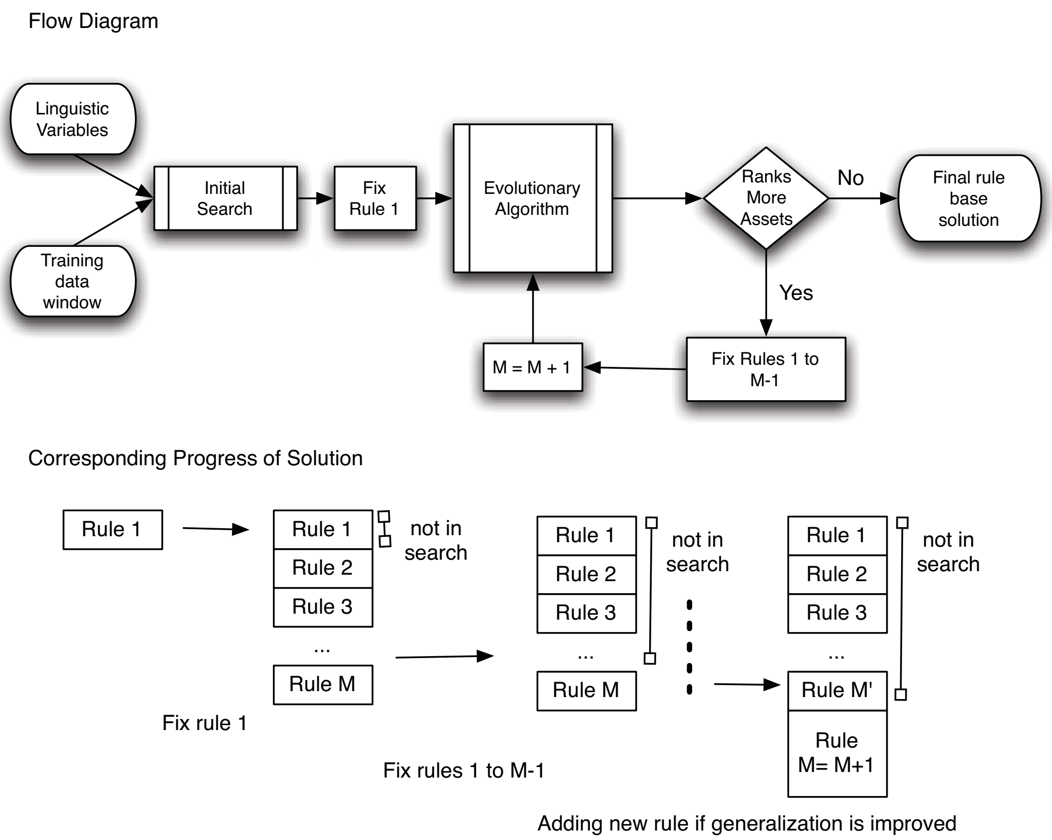


Figure 8.4: The construction of a single rule base.

3. Apply r-SAME to the whole population using a previous searches best rule base and double parameter $p \in [0, 1]$ if available

4. Evaluate each solution: calculate $eval(\rho_v)$ for $v = 1, \ldots, n$

5. Identify the best solution,$cBEST$ in P

6. If $eval(cBEST) > eval(BEST)$ then $BEST = cBEST$ and $SWI = 0$

7. Alter the population by applying a mutation and crossover operators (tournament selection of size 2 is used)

8. Apply repair operator to each offspring to fix illegal variations with respect to the best solution $best_{previous}$ from the previous generation (elitism is not used)

9. If parameter to use fixed rules is used, apply r-FIXED operator to each offspring and a single global fixed rule base

10. If parameter is set apply r-SAME operator to each offspring using the current *best*

11. $generation = generation + 1$

12. $SWI = SWI + 1$

13. Repeat steps 3 – 12 successively until SWI is $maxSWI$ and no improvement is recorded

14. If $generalization(BEST) > generalization(FIXED)$ set $FIXED = BEST$, $generation = 0$, $rules_{max} = rules_{max} + 1$ and return to step 3; else return P

where *generalization* is a function to return a value that measures the generality of solutions, a number of choices are possible and in the experiments discussed here we defined it as the number of stocks that obtain a rating on application of a rule base. Internally, rulebase solutions are represented using arrays of floating point, integer and boolean values and altered using several standard cross over and mutation operators. This representation is given in detail in in Chapter 4 and operators and their adaptation is described in Chapter 6.

The fitness of individuals is measured by examining the ability of a rule base to correctly rank stocks in comparison with an ideal rank. Figure 8.5 shows a visualization of the evolutionary process where the correctly ranked stocks improves as the process progresses. The figure shows the improvement of in best fitness rule base stock selection ability during the learning process. Each tile shows the top 10% of assets in a ranking obtained by ordering by rulebase output rating. A green square indicates a rulebase implied rank that in the top 20%. A red square means a placement in the bottom 50% of actual relative performance of the pool of possible asset choices over the training window.

8.6 Aggregation of Rule Bases to Construct a Recommendation

The objective of the optimization process for rule bases is controlled by changing the method to obtain the ideal ranking used for evaluation. We take an approach of using solutions for specific tasks. Separate rule bases are optimized to predict increasing and declining asset price performance as well as asset volatility over *both* long and short forecast horizons. The method for aggregation acts on the rankings from rule base, i.e. the application or phenotype level.

To combine rankings $A^{\rho_{x_1}}$ and $B^{\rho_{x_2}}$ from rule bases ρ_{x_1} and ρ_{x_2} that have fitness x_1 and x_2 we require that each implies an ordering of a set of listed assets $M = \{s_1, s_2, \ldots, s_m\}$. A new ranking with input from each solution is found by ordering the set by a rating:

$$\frac{x_1 a_{s_{c_i}} + x_2 b_{s_{c_i}}}{x_1 + x_2}.$$

To obtain the negative (used in Figure 8.6) the difference is taken. Figure 8.6 illustrates the components of the buy and sell recommendations.

As new data is loaded the prediction model is updates itself by repeating the search process and linguistic variable specifications. To apply the recommendation a simple management routine was used as follows to update a portfolio once a month. A managed portfolio contains up to 40 assets held with either long or short positions but not both. Only five stocks can be bought or shorted in any update to reduce the risk from investing a large amount of capital based on a single solution. To promote portfolio diversity, no stock can be bought or shorted if it is already held. Highly rated assets are bought (and if any short positions in these are also closed) and those ranked highly by the sell recommendation are sold (or short positions are opened). In the absence of new decision recommendations from the prediction model any stocks in the portfolio are sold after being held for six months. In addition, the system allocates available cash depending on the fitness of solutions used to construct the buy and sell recommendations in training.

8.7 Experimentation

This section provides experimental results testing the system. We use a universe of stocks comprising the ASX 200 Index. Data was sourced from DataStream International (http://www.datastream .com/). Transaction costs of 0.5% we used in the simulation, all price data was adjusted for unusual events such as stock splits, new issues and similar factors, trading was only possible in a stock if its trading volume was greater than 0.

In earlier work we tested an evolving fuzzy rule portfolio management system for long only trading (i.e. not a hedge fund scenario with short selling as presented here) using a limited subset of inputs derived from price and volume data to trade the same Index [36] and the MSCI Europe Index [37] for the period 2000 to 2006 and 1991 to 2005 respectively. It was found that this adapting fuzzy rule based technical trading system achieved an average annual return of 34% [36]. In this paper we partly continue this work according to suggestions of earlier reviews and test an enhanced system with some common basis with additional model factors during the subsequent period August 2005 to January 2009 in challenging market conditions. From November 2007 at the time of the sub-prime mortgage crisis in the US and onwards the market was trending negatively as the liquidity crisis unfolded, see Figure 8.7.

The goal of the experiments is primarily to examine the possibility of improvements in portfolio management performance from using the methodology. A secondary objective is to investigate the behavior of the system in its method of selecting stocks.

Table 8.2 provides performance results describing the return and volatility of managed portfolios in comparison to the index, random stock selection, and without using solution aggregation. The portfolio that did not use the method of aggregating solutions was constructed rankings obtained directly from the output of two rule bases – one for buying and another for selling. Figure 8.7 shows return on investment over time for simulated funds invested using the system and comparison portfolios. Figures 8.10 and 8.8 shows the performance of individual assets and industry sector components of a managed portfolio to give an insight into the stock selection behavior and diversity over different types of stock that is achieved. In general a diversified portfolio less risky and robust in market downturns.

It is clear that the portfolios managed using our approach outperformed all comparisons over the period and on an annualized basis. The annualized return was 30%, a value almost equivalent to performance during previous tests in a rising market using a long strategy. In the test of *not* combining solutions return performance was 12.8% per annum. The random selection method and index experience loss -20% and -3% respectively. It was the case the managed portfolio experienced a significant fall in value during the financial crisis. However, in percentage terms the loss was around half that recorded by the market index (see Figure 8.7).

Volatility is defined as the standard deviation of daily log returns. It is in essence a measure of risk [98] and it is better to show a lower portfolio volatility. Table 8.2 shows the volatility of all the portfolios tested. The aggregated solution portfolio was less volatile in its return over time than the non aggregated approach approach and the random portfolio. This shows that there seems to be some advantage in explicitly optimizing solutions for this objective for return as well because the non aggregated portfolio only targeted returns, however it also did not outperform on either measure.

The Sharpe ratio [98] is a ratio of the excess return of the portfolio above the risk free interest rate and its annualized volatility. Its meaning is an indication of how well an investor is compensated for volatility or risk by the returns obtained. The system was able to achieve significantly better risk adjusted returns by this measure than the comparison portfolios (except the previous test of the approach in a rising market).

Figures 8.10, 8.8, 8.11 and 8.9 shows the performance of individual stocks selected within a portfolio and by industry sector. A limited number of stocks are bought or short sold each month and held for a fixed period in the absence of further ratings. A large contribution can be made by single excellent selections: for example the purple series seen in Figure 8.10 caused the portfolio to double in value. The benefit of short selling are for hedging in a falling market are observed in the second half of the test, short positions positive returns that reduce the impact of the falls experience by long components. Notably, the graphs show long positions in the basic materials sector contribute substantially to growth for most of the period. This segment of the Australian economy grew dramatically in a resources boom, the system was able to allocate funds to take advantage of this. The financial sector fell in a global liquidity crisis in the second half of 2008. The system was able to profit from short selling financial stocks. It is also the case that during time when most listed stocks were increasing in price the system selected stocks that did not cause large losses and then was able to gradually increase short exposure as market conditions worsened using information from model training performance.

Variable #	Factor Description
f_1	Jensen's α (calculated using previous 120 trading days).
f_2	β (previous 120 trading days).
f_3	Dividend yield for stock over sector average.
f_4	Change in dividend yield for stock over sector average.
f_5	Price to book value over sector average.
f_6	Change in price to book value over sector average.
f_7	Price earnings ratio over sector average.
f_8	Change in price earnings ratio over sector average.
f_9	Forecast of price earnings ratio for the next year by financial analysts over sector average
f_{10}	Change in forecast of price earnings ratio over industry average.
f_{11}	The market capitalization of a company.
f_{12}	Change in the market capitalization of company.
f_{13}	Earnings per share over sector average.
f_{14}	Change in earnings per share over sector average.
f_{15}	Total debt to equity ratio over sector average.
f_{16}	Change in total debt to equity ratio over sector average.
f_{17}	Long term debt to equity ratio ($>$ 1 year) over sector average.
f_{18}	Change in long term debt to equity ratio ($>$ 1 year) over sector average.
f_{19}	Earnings before interest and tax over sector average.
f_{20}	Change in earnings before interest and tax over sector average.
f_{21}	Return on assets over sector average.
f_{22}	Change in return on assets over sector average.
f_{23}	Return on equity over sector average.
f_{24}	Change in return on equity over sector average.
f_{25}	Money flow index.
f_{26}	Near term price change (3 month).
f_{27}	Long term price change (1 year).
f_{28}	Bollinger bands.
f_{29}	Volatility (standard deviation for previous 3 months).
f_{30}	Price volume oscillator.

Table 8.1: Model factors, each factor modeled by a linguistic variable.

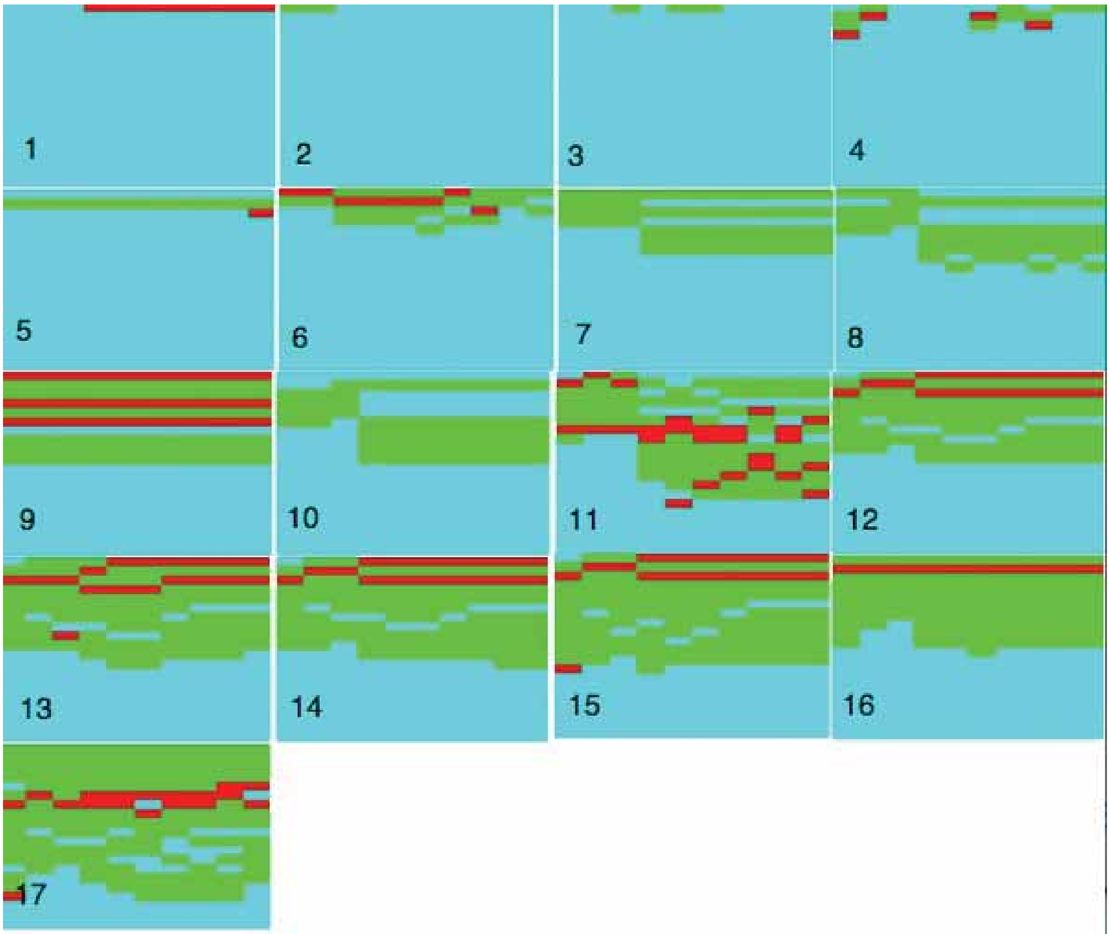


Figure 8.5: Visualization of the change in fitness of a rulebase implied asset ranking during an evolution process. The meaning of a single tile is shown in Figure 8.3.

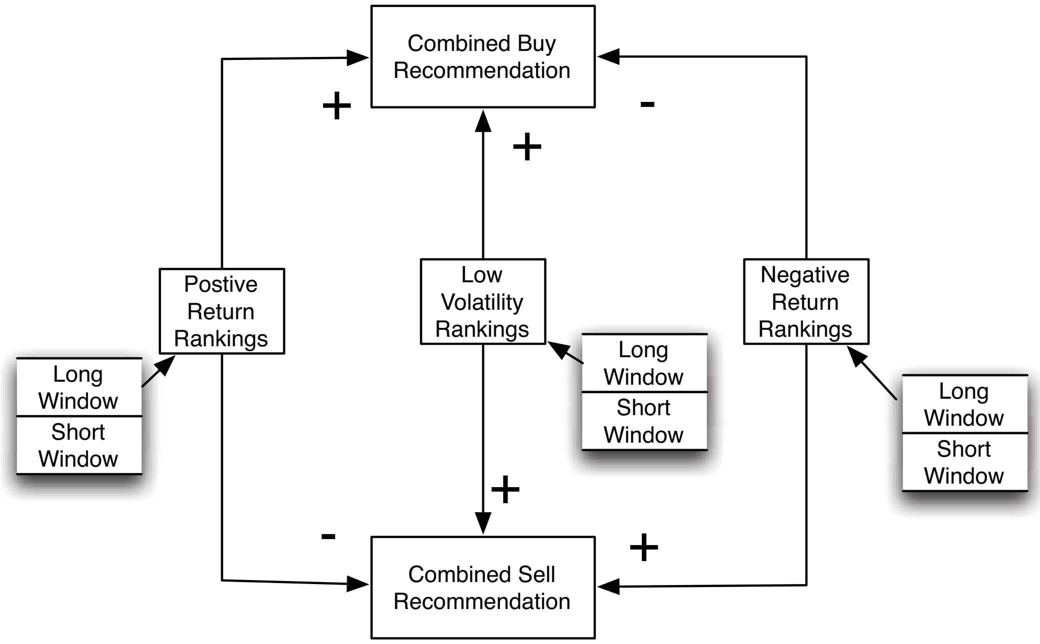


Figure 8.6: The aggregation of multiple solutions in constructing the recommendation.

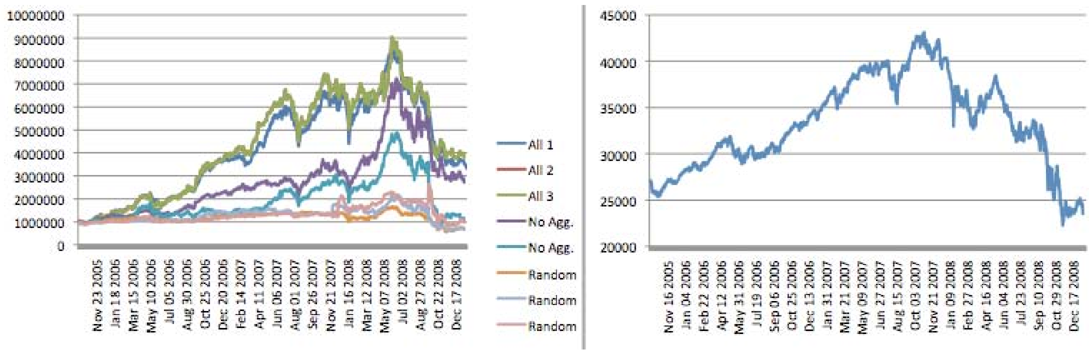


Figure 8.7: The value of portfolios managed in simulation during the experiments (left) and the index (right).

Metric	Index	Previous Test ('00-'06)	Random Sel.	No Sol. Agg.	Sol. Agg.
Hp. ret.	-0.100	na	0.059	0.615	1.36
An. ret.	-0.033	0.321	-0.199	0.128	0.307
An. vol.	0.200	0.205	0.365	0.492	0.360
Sh. ratio	-0.491	1.404	-0.358	0.179	0.67

Table 8.2: Performance statistics for return and volatility in portfolios tested and comparison with the market index as well as tests provided in Chapter 6 for a long only portfolio traded in a rising market from 2000 to 2006. Values are averages from three simulation runs.

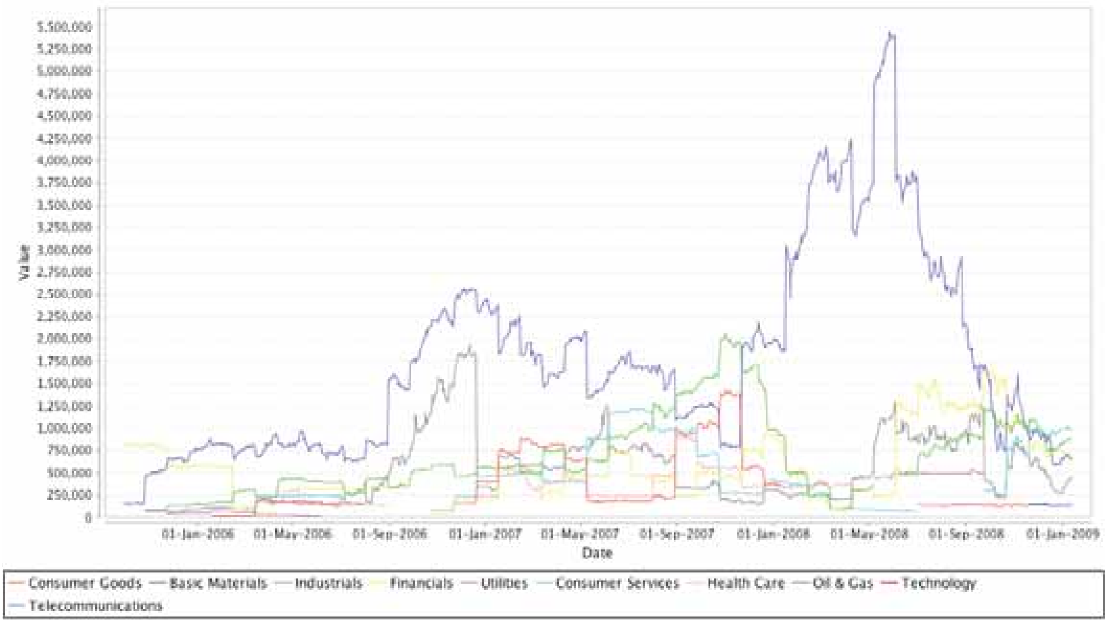


Figure 8.8: Value of long positions in a single portfolio by sector.

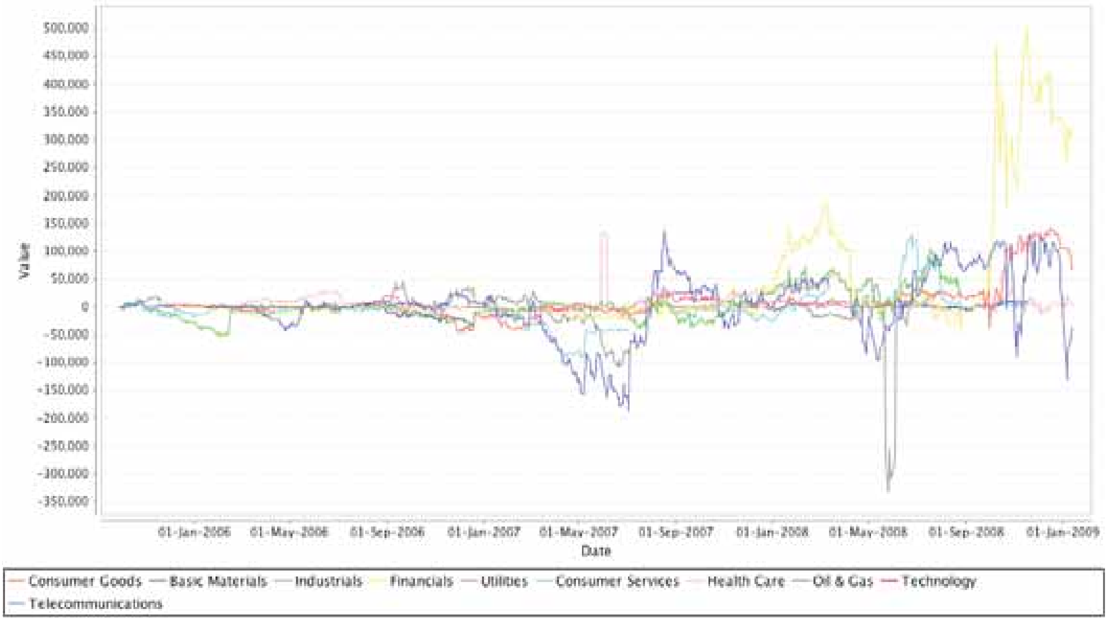


Figure 8.9: Profit from short positions in a single portfolio by sector.

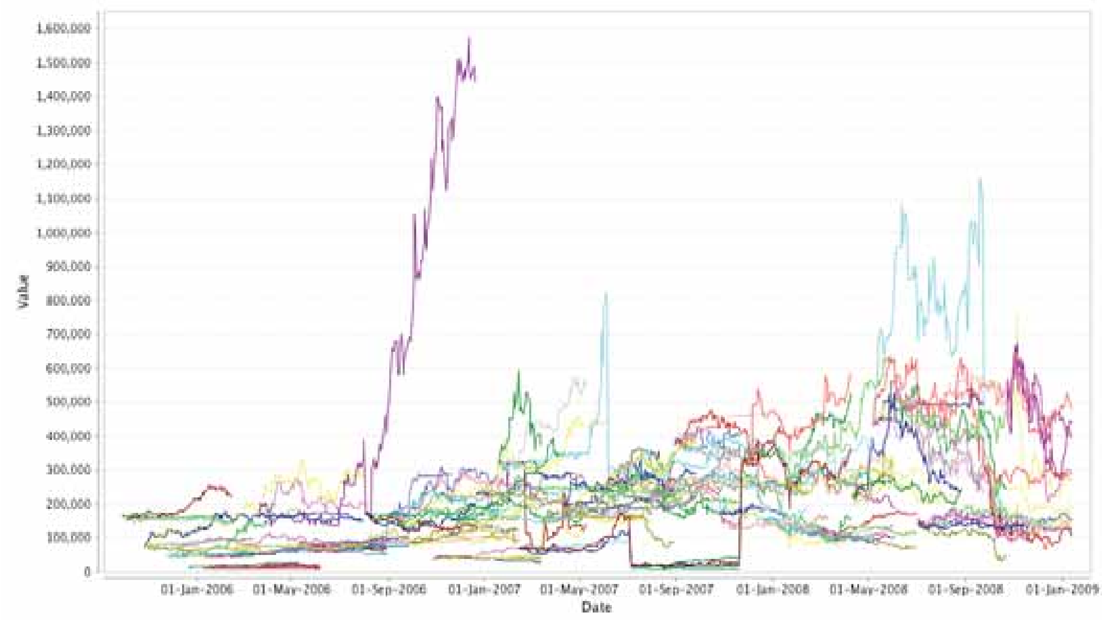


Figure 8.10: Value of long positions in a single portfolio. Each series shows the change in value of a single component (stock) that forms part of a portfolio.

Figure 8.11: Profit from short positions in a single portfolio. Each series shows the change in value of a single component (stock) that forms part of a portfolio.

Chapter 9

Decision Support System

This chapter provides a user application for adaptive quantitative fund management making use of the technology provided in the earlier chapters. The system is an adaptive intelligent decision support system capable of handling large volumes of data for automatic opportunity assessment, evaluation of sector trends and prediction of future opportunities through adaptive learning. It is useful as a tool to assist investment decisions by an analyst or as a stand alone comprehensive methodology for asset allocation and management.

We include here an overview of use cases (including back testing, performance analysis, report generation and essential management processes), as well as description of the user interface and discussion of business models suitable for applying the system in context of the fund management industry.

9.1 Use Cases

This section provides a summary of the portfolio management activities that are able to be carried using the software. These tasks fall into five main categories: model tuning; periodic strategy updates and portfolio rebalancing; reporting; incorporating user knowledge and external outlooks; and market analysis. Let us discuss each of these separately in the following subsections.

9.1.1 Model development and back testing

The primary purpose of the system is to develop and apply trading strategies. A model (trading strategy) is a specification of explanatory variables, the information set, and relationships between them together with an asset valuation meaning given particular expressions of these. The user is able to select input data types and derivations, these from the information set. The magnitude of these inputs is controlled using fuzzy membership functions with a natural language interpretation understandable by the user. Using linguistic descriptions of the information set Îf-then causal sentences are able to be expressed, fuzzy rule bases (see Chapter 4).

In order to determine a models suitability, the user is able use the software package to do the following:

- Specify trading strategies manually.
- Allow the system to learn models from past data using particular data windows, model parameters, and algorithms.
- A combination of these two.

Essentially, the user is able to test the performance of potential models using past data and measure in detail the performance that would have been obtained. This is termed back testing and involves, in the case of algorithm developed strategies, out of sample tests simulating real trading in as detailed a manner as possible by including transaction costs, cash interest rates and so on. A detailed set of standard performance analytics are provided to interpret these back tests, see section 9.1.3 below. Once a suitable model or specification for model learning is obtained these settings are able to be used in portfolio management.

9.1.2 Portfolio construction and management

The management approach implemented in the system is based around periodically updating a portfolio on the basis of the learned recommendation, termed portfolio rebalancing. This takes place at set intervals, for example each month or each week, although the interval can be varied if the user would like. The process serves two functions: to maintain the portfolio within constraints specific to a management style such as sector exposure, maximum allocation to particular assets, long short ratios, index weightings and the like; and to readjust the portfolio contents depending on a changing view of the market from the input information set and the asset valuation strategy (which also evolves over time).

Three management styles are provided that utilize the software:

- Long Only,
- Long/Short,
- Enhanced Index Tracker.

The Long Only style involves recommendations to buy assets and hold for a set period unless a sell recommendation occurs. The Long/Short style involves investment vehicles that utilize both long positions and short selling. A facility is also provided to manage an Enhanced Index Tracking portfolio is managed to (1) track the market index by holding all stocks in the index in the same proportion to their index weight, and (2) adjust the weighting for some set number of positions depending on the strategy recommendation so that the portfolio is under or over weighted in these to target a slightly superior performance to the index with a smaller risk of catastrophic failure (note the superior performance is expected to be more modest as well). The risk and return preferences of the investor as well as available cash and willingness to bear transaction costs would inform the decision to select different management styles.

Depending on the style, the system provides a recommendation to buy, sell, short or hold assets in the portfolio from one rebalancing period to the next. The configuration allows the user to specify parameters that influence the recommendation (to further control how a strategy is applied and the risk) including the rebalancing period, the maximum cash to bet on rules, the ratio between long and short allocations, the maximum deviation from the index weighting, the number of stocks to hold, the sector weighting allowable and others in the configuration. The recommendation comprises of a number of recommended transactions or the recommended stocks from the strategy with a numerical value to indicate strength of recommendation.

Performance analytics are provided to follow the performance over time, these are discussed in the next section.

9.1.3 Reporting and performance tracking

Portfolio performance measurement involves comparison and interpretation of performance statistics, in particular relating return to the risk free rate and an index benchmark to quantitively determine management performance in terms of risk and return. Detailed descriptions of calculating these measures and their application and interpretation are given in Chapter 5. These statistics are benchmarks in the industry used to compare investment vehicles and measure success.

An important distinguishing point fundamental to the approach we have used is that the specification of the models using fuzzy logic rule bases. Unlike many comparable approaches (for example neural networks or grammatical evolution which are basically black boxes as far as being interpreted by humans) the strategy specification itself is readily understood and interpreted: this means it is feasible to justify or check decisions in managing a portfolio using the software by reporting the information set used and the interpretation of this information.

9.1.4 Incorporating analyst knowledge and strategies

Rather than using a strategy completely dependent on the learning algorithm and historic data window, a user may prefer to specify some part of the strategy to be used. This may be to minimize risk or to promote other properties in a portfolio (for instance, high alpha is a common goal for portfolio managers). In addition, an analyst user is able to specify that the portfolio should be constructed from a subset of the universe of stocks or from specific industry sectors. One possibility is for an investment product that holds only mining and resource stocks or some other asset class. The information set can be actively varied by the user, for example to be restricted to specific inputs or whose use are justified by research or possibly legislation or risk policies.

Incorporation of expert knowledge about strategy formulation is implemented by extending the fuzzy logic rule base specifications. A user specifies a rule base (for instance that some measure of risk such as beta is low, which may be defined as a member of the lowest percentile of observations depending on the number of membership function parameter. This specification can then be combined with a learned strategy in two ways:

- An exclusion or inclusion filter which has the meaning all items in the portfolio must not or must fit the criteria,

- To be combined by boosting the recommendation using a fuzzy AND operator to combine the output of the learned strategy and the user specified rule base.

9.2 Interfaces and User Interaction

In this section we provide an overview of the software package, functionality and user interaction. The application is written in Java (Version 6). The package is hence platform independent. There is a requirement for at least 1.5 Gigabytes of RAM to run.

9.2.1 Data

Data is initially loaded into an internal data base from specially formatted comma separated value files. The data is in a precise format where each row corresponds to a date and each column series relating to an asset. Global data types (not specific to particular assets) are in *date, value* format.

Figure 9.1 shows the asset listing screen which shows all assets loaded into the system data base. On selecting a stock its relevant series may be view as shown in Figure 9.2. Similarly the available derivations (linguistic variables) are also able to be viewed for historic periods. The input data screens also show global data types in a separate tab.

Viewing these screens emphasizes the impossibility of interpreting vast quantities of raw data values. However, these screens are useful for error checking and simulation verification by comparison with portfolio results.

9.2.2 Data visualization

The most common method used in practice to interpret data series by users is charting. A plethora of software packages are based around charting and visualization to identify opportunities in large quantities of data. Figure 9.3 shows the screen which charts the risk free rate and index global data types which give an overview of changing market conditions in the historic data available for training or simulation runs.

Figure 9.4 shows the charting of stock price and volume series together with technical and other indicator series over the test period. The "Combo box" in the

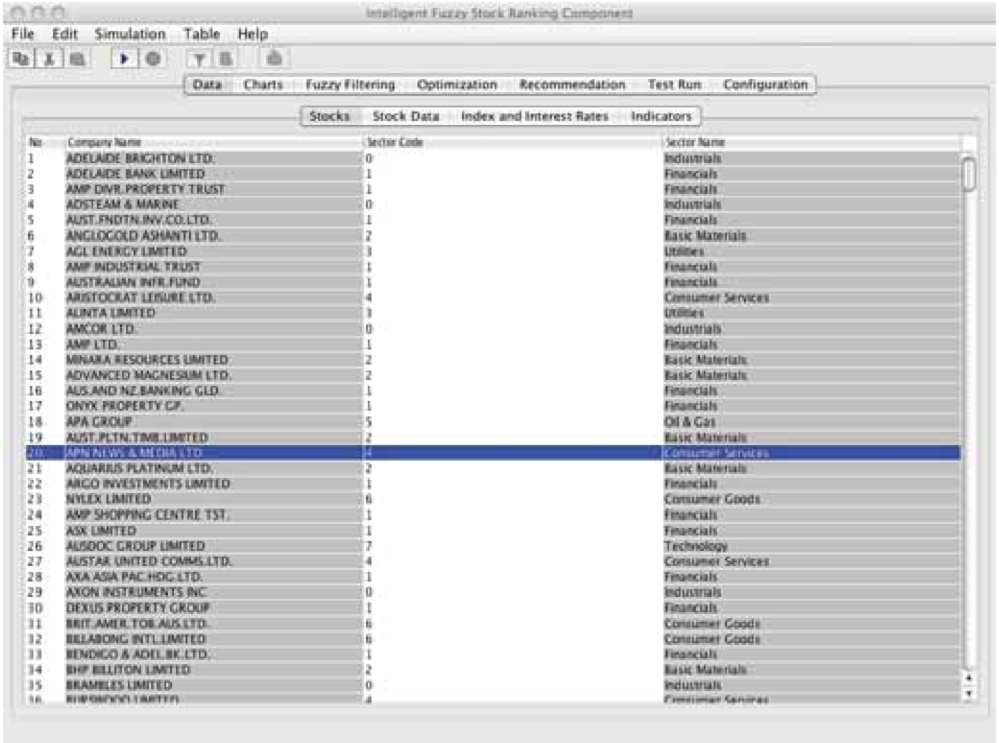


Figure 9.1: Stock screen.

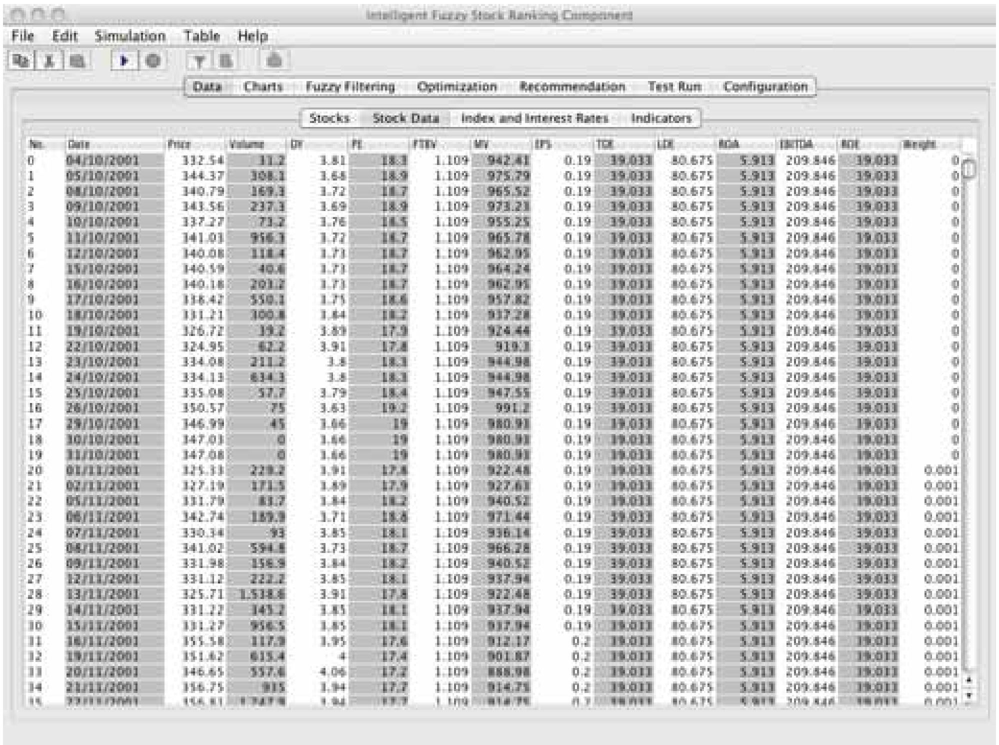


Figure 9.2: Stock data screen.

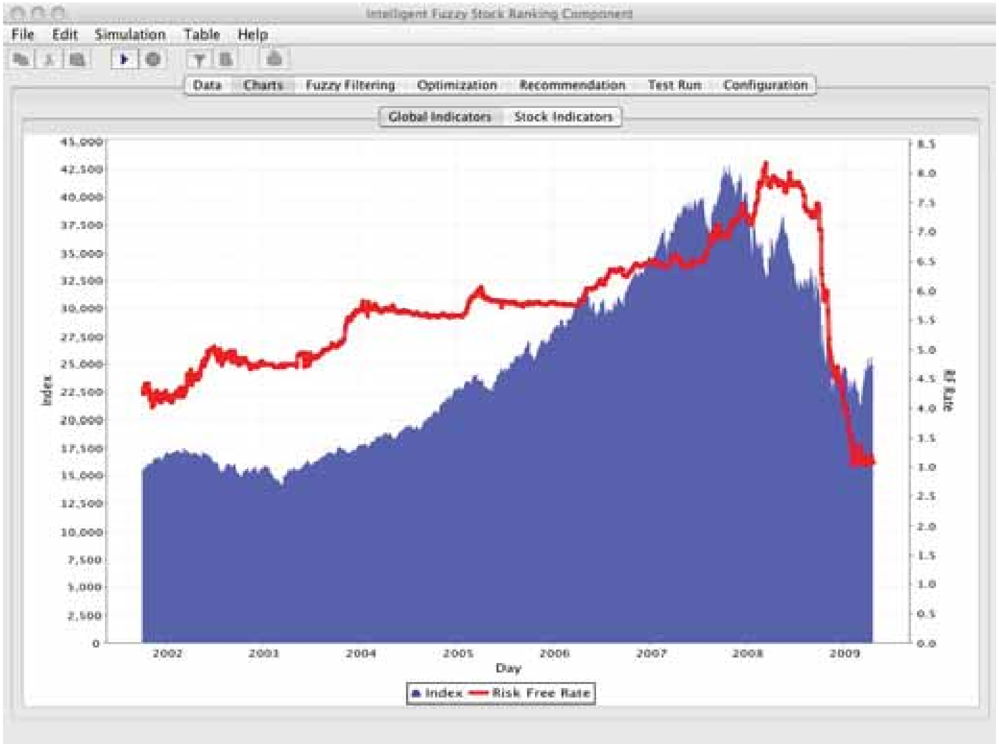


Figure 9.3: Global data screen.

top right hand corner of the tab allows selection of indicators to plot; the list on the left side of the allows selection of an asset. For each asset the indicators correspond to the fundamental and technical linguistic variable factors discussed in Chapter 6. A functionality is provided for zooming as well as printing all charts to "png" image files.

As with the raw data screen, the limitations of charting as a tool for interpreting large volumes of data are emphasized in comparison with the intelligent automation of the learning approach in which a type of reasoning about the observed data takes place. To interpret and form accurate relative conclusions using graphical tools would be prohibitively time consuming task. In practice, an analyst would generally focus on a subset of assets and analysis techniques and information they are familiar with despite potentially having access to a huge information set. Again, the primary use of this tab is data verification and performance checking.

In a demonstration this tab serves to emphasize the substantial benefit that

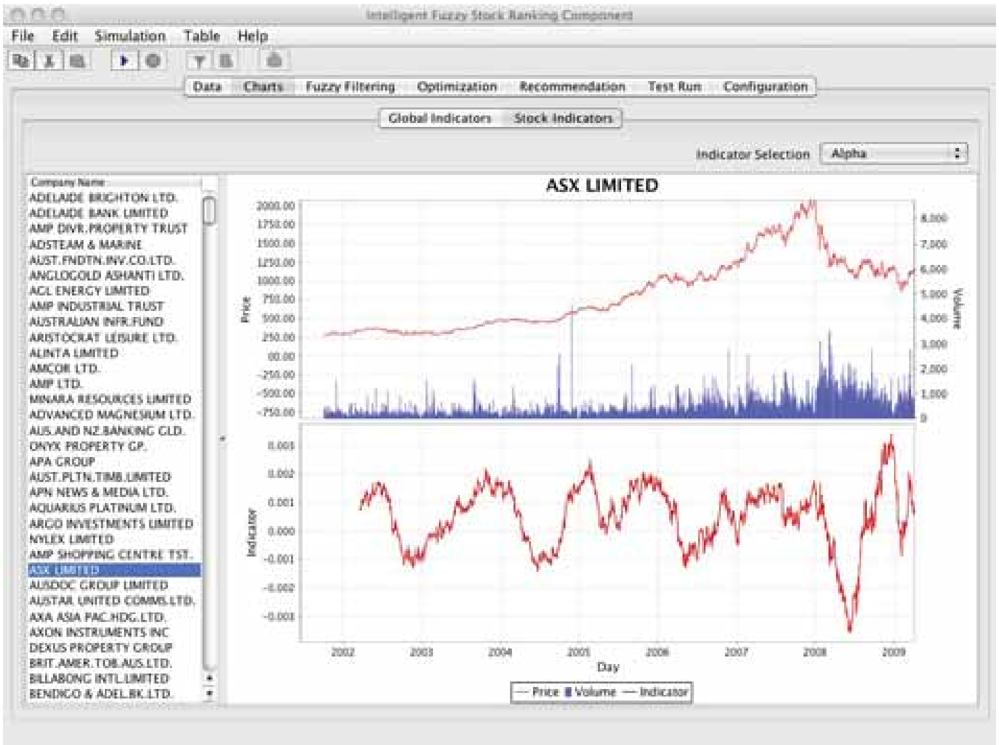


Figure 9.4: Indicators screen.

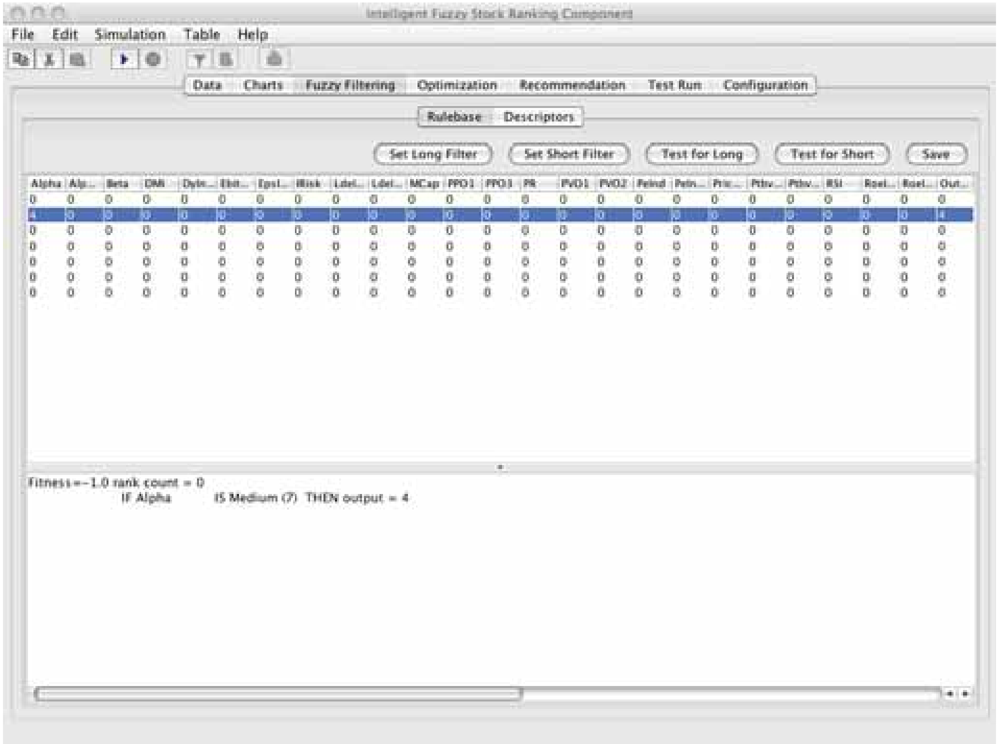


Figure 9.5: Asset filters screen.

computation provides in interpreting data rather than just presenting knowledge (as is traditionally the case in business information systems). For example, it is a possibility to demonstrate visually that particular inputs have been useful in the past for predicting price movements. And possibly with a small improvement to use a search to identify find on average the most successful historically. However, the novel and significant contribution this software makes is in providing a recommended course of action or decision based on interpreting this knowledge and in being able to update the decision models.

9.2.3 User strategy construction

Figures 9.5 and 9.6 show the user interface for constructing manual rule base definition. This is useful for inserting application domain knowledge or generating comparisons or factor analysis (possibly to limit a later search to only consider fac-

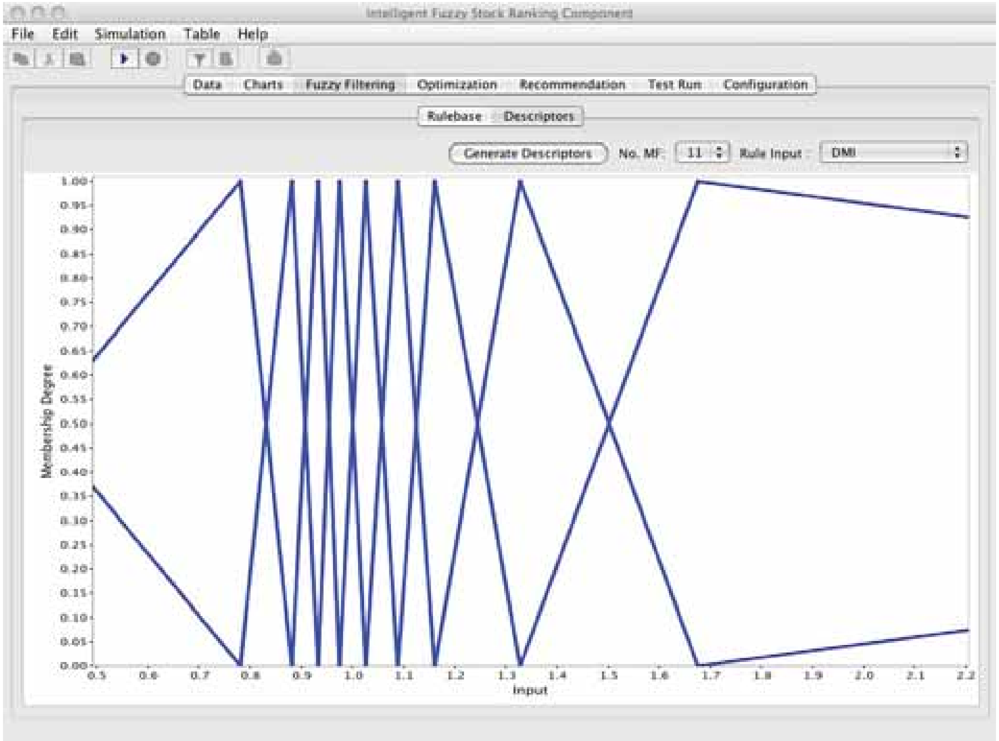


Figure 9.6: User membership function specification.

Figure 9.7: Optimization start button in the toolbar.

tors with at least some relevance). Using the table in the top of Figure 9.5 the user is able to interact directly with the rule base specification by setting membership levels required for the linguistic variables in a rule base. The output level is also able to be set. In this way quite complex structures are able to be used to influence the management behavior, it is possible to make a requirement that all stocks in the portfolio have a minimal level of alpha, or a minimal market capitalization for instance. User generated rules are also able to be used as seeds for the search process.

Depending on the specification for the number of membership functions and the period for updating them the fuzzy rules default to a much simpler trading method in which assets are ordered by magnitude of a factor (when the rule base is applied for ranking), this corresponds to classical financial portfolio generation techniques where, say, the top percentile of stocks by a particular factor are held each rebalancing period.

9.2.4 Strategy optimization

To start an optimization process the user is able to click on the start button in the tool bar control panel (see figure 9.7). In addition a user is able to run a simulation of using a particular optimization configuration, or possibly a fixed strategy, in a particular historic data period (see Figure 9.8).

Figure 9.9 shows the evolving rule base during a search process: the left panel

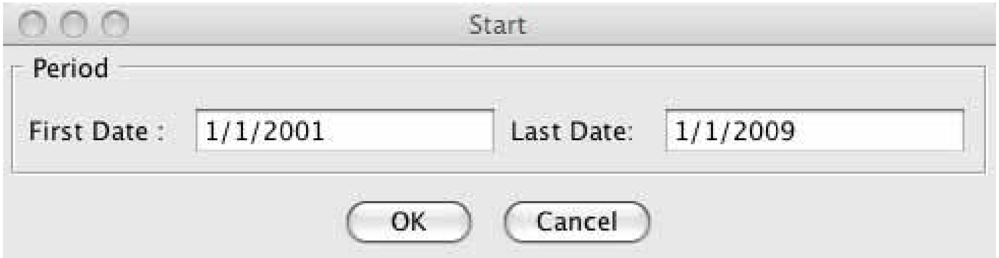


Figure 9.8: Simulation start dialog.

lists the best rule base found at progressive generations and the right hand panels show the membership functions (meaning of linguistic descriptions, top right panel) used in the rule base (bottom right panel). Figure 9.10 shows the rankings produced by the best rule base strategy selected in the left panel: correctly selected (according to the fitness objective) stocks are marked green. A successfully evolved strategy would have all the highly ranked stocks marked in this way.

These screens enable a user to understand the progress of the optimization process and alter parameters to achieve desired objectives. For instance avoiding premature convergence, maintaining a balance between model simplicity and fidelity (generalization), and other aspects of strategies that are generated such as interpretability.

9.2.5 Recommendation

Using an evolved rule base the system is able to construct a recommendation to buy or sell stocks or to carry out transactions. Figure 9.11 shows the ranking recommendation of assets relative to one another from applying a solution (either a single rule base or an aggregate solution). Figure 9.12 shows transaction recommendations for a portfolio used to rebalance given a particular portfolio managed over time. Each time an optimization process is completed a separate set of transactions are recommended for the user to implement.

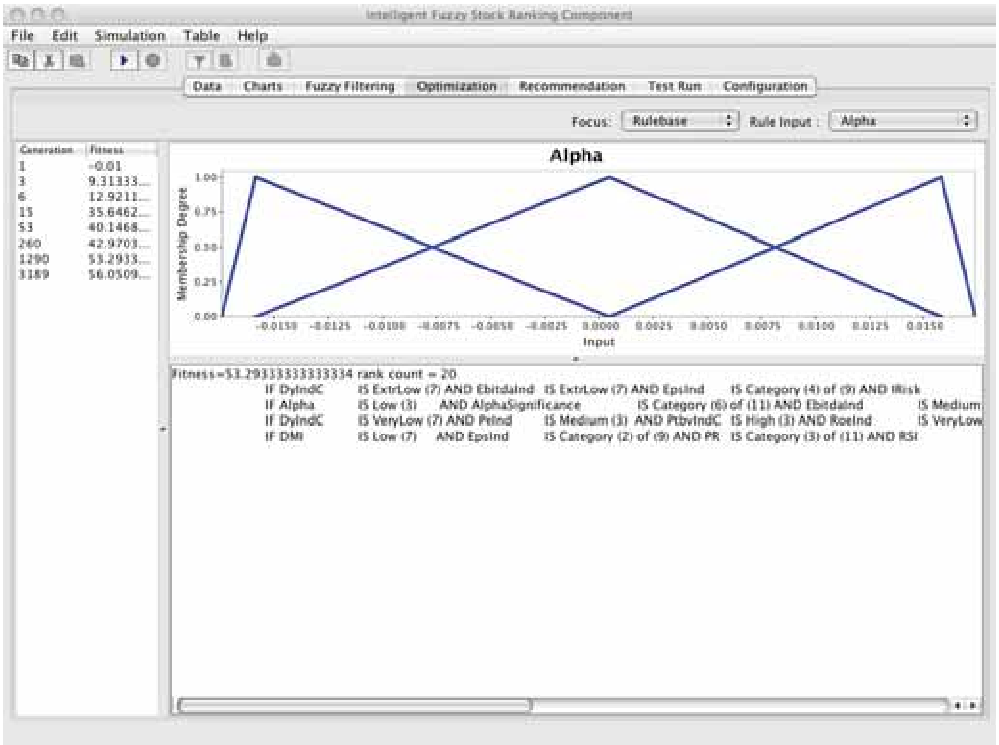


Figure 9.9: Optimization screen.

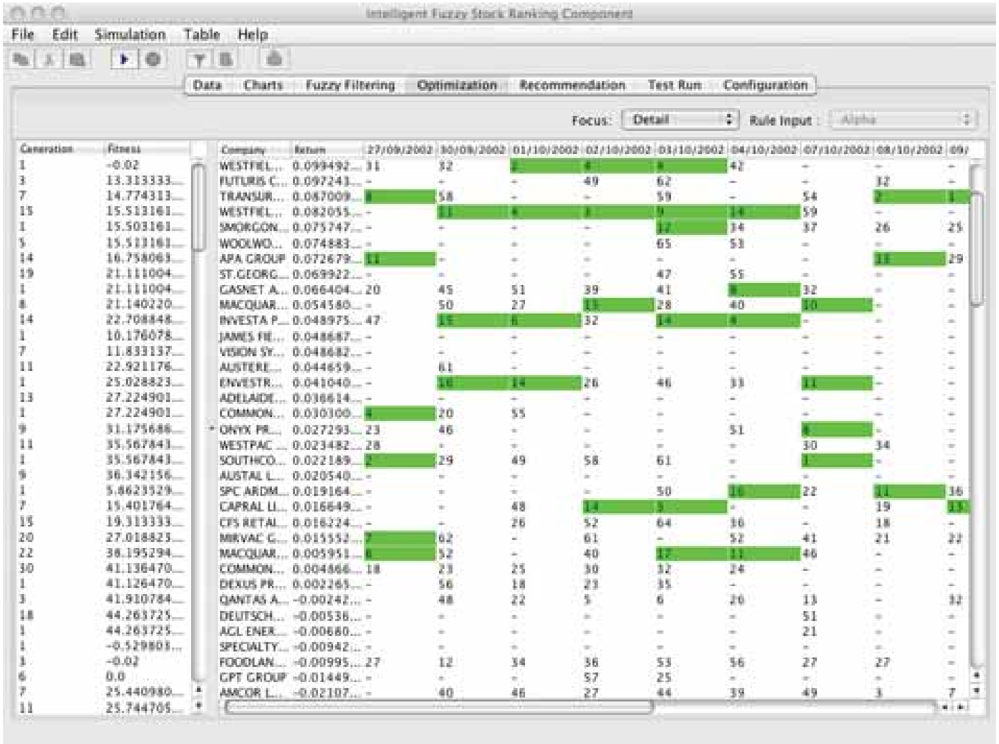


Figure 9.10: Detailed view of evolved strategy fitness in selecting top stocks in the training data sample.

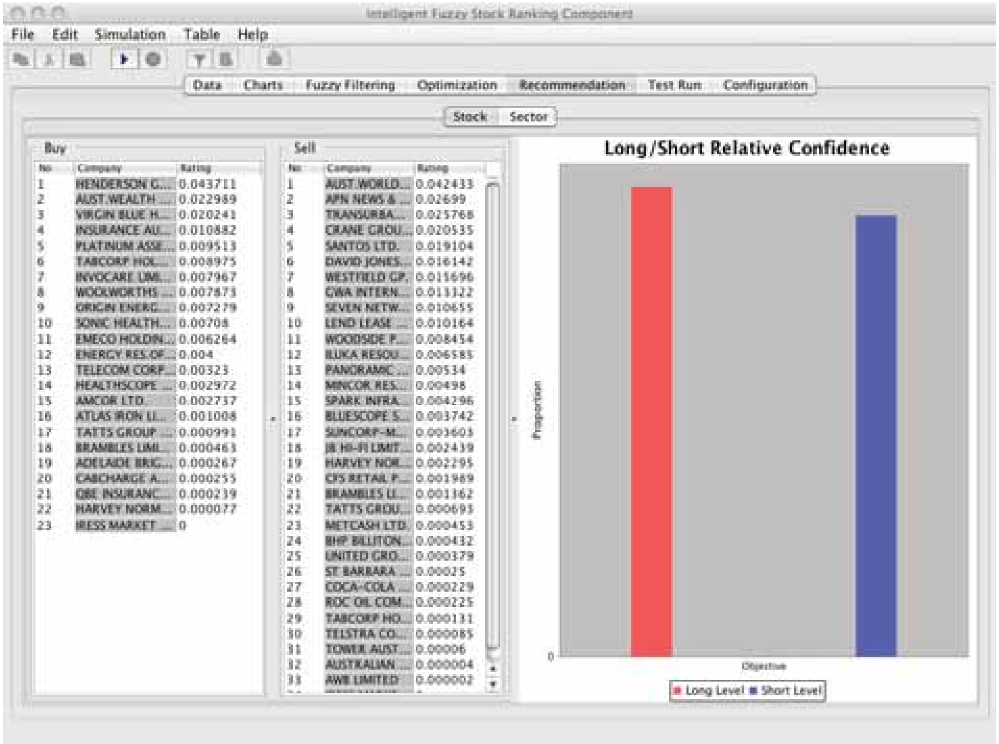
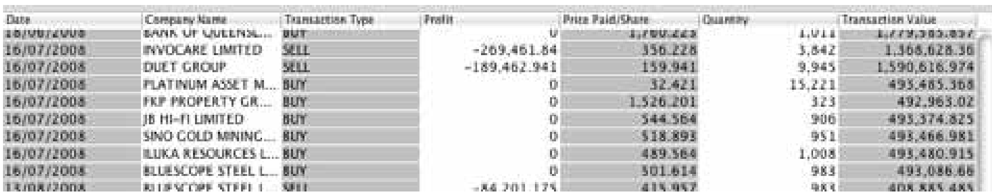


Figure 9.11: Recommendations screen including stock ranking and long short relative confidence (calculated from the rule base fitness in aggregate solutions).



Date	Company Name	Transaction Type	Profit	Price Paid/Share	Quantity	Transaction Value
[illegible]	BANK OF QUEENSL...	BUY	0	[illegible]	1,011	[illegible]
16/07/2008	INVOCARE LIMITED	SELL	-269,461.84	356.228	3,842	1,368,628.36
16/07/2008	DUET GROUP	SELL	-189,462.941	159.941	9,945	1,590,616.974
16/07/2008	PLATINUM ASSET M...	BUY	0	32.421	15,221	493,485.368
16/07/2008	FKP PROPERTY GR...	BUY	0	1,526.201	323	492,963.02
16/07/2008	JB HI-FI LIMITED	BUY	0	544.564	906	493,374.825
16/07/2008	SINO GOLD MINING...	BUY	0	518.893	951	493,466.981
16/07/2008	ILUKA RESOURCES L...	BUY	0	489.564	1,008	493,480.915
16/07/2008	BLUESCOPE STEEL L...	BUY	0	501.614	983	493,086.66
13/08/2008	BLUESCOPE STEEL L...	SELL	-84,201.175	415.957	983	408,885.485

Figure 9.12: Recommendations screen — transaction recommendations.

9.2.6 Performance analysis

An extensive set of standard performance measures and reporting capabilities are provided. These are used for analyzing performance in historical simulations and as the system is applied for fund management. The following statistics are provided (see Figure 9.13:

- Holding period return,
- Annualized arithmetic return,
- Annual geometric return,
- Annual volatility,
- Excess return over RF rate,
- Capital Asset Pricing Model regression: alpha, beta and R^2,
- Information ratio,
- Annual expected return,
- Selectivity,
- Net selectivity,
- Churn (annual cash turn over) to show transaction costs.

Rolling statistics of portfolio alpha and daily return provided in chart form to show the performance change over time. In the case of the Enhanced Index Tracking portfolio style an additional statistic termed tracking error is provided to measure the deviation from the index, this is essentially interpreted as a measure of risk. Examples of these charts are shown in Figure 9.14.

In addition to these statistics a number of reports are able to be printed to indicate management behavior over time. These include:

- Previous transactions with those as a result of the strategy highlighted (as opposed to due to constraints or standard processes),
- Return over the portfolio life compared to the index and interest rate,

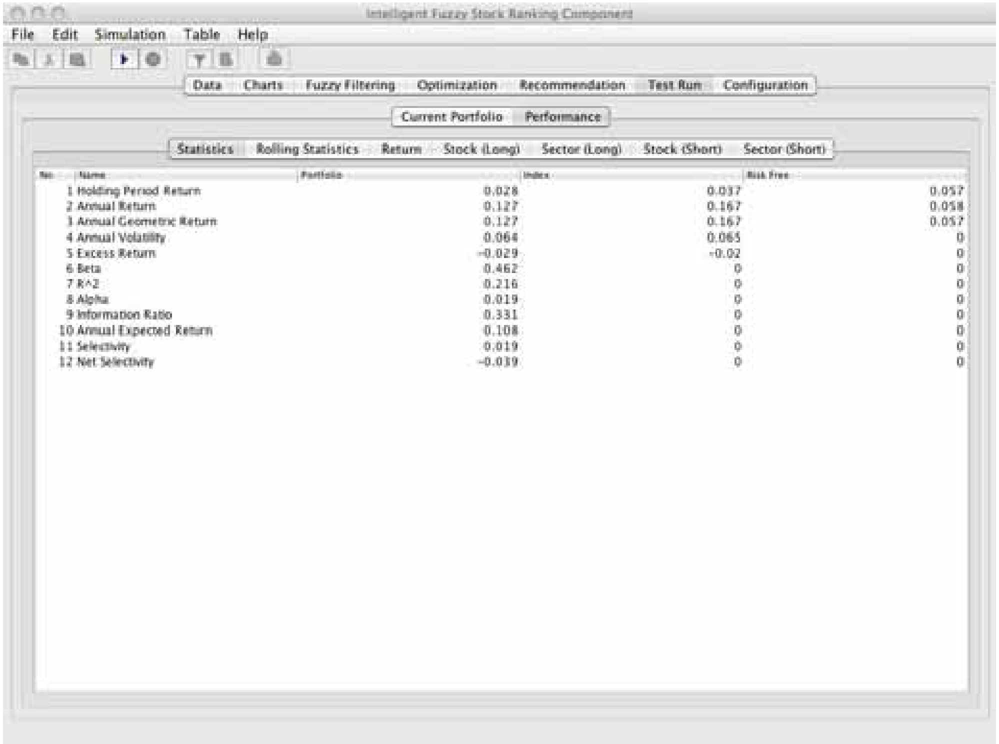


Figure 9.13: Performance statistics screen.

- Visualization of individual profitability of portfolio components (Figure 9.15),
- Sector allocation of the portfolio (Figure 9.16),
- Index and interest rate change over time,
- Information set plotting,
- Strategy specifications used for trading over the life of the portfolio,
- Current portfolio positions (Figure 9.17).

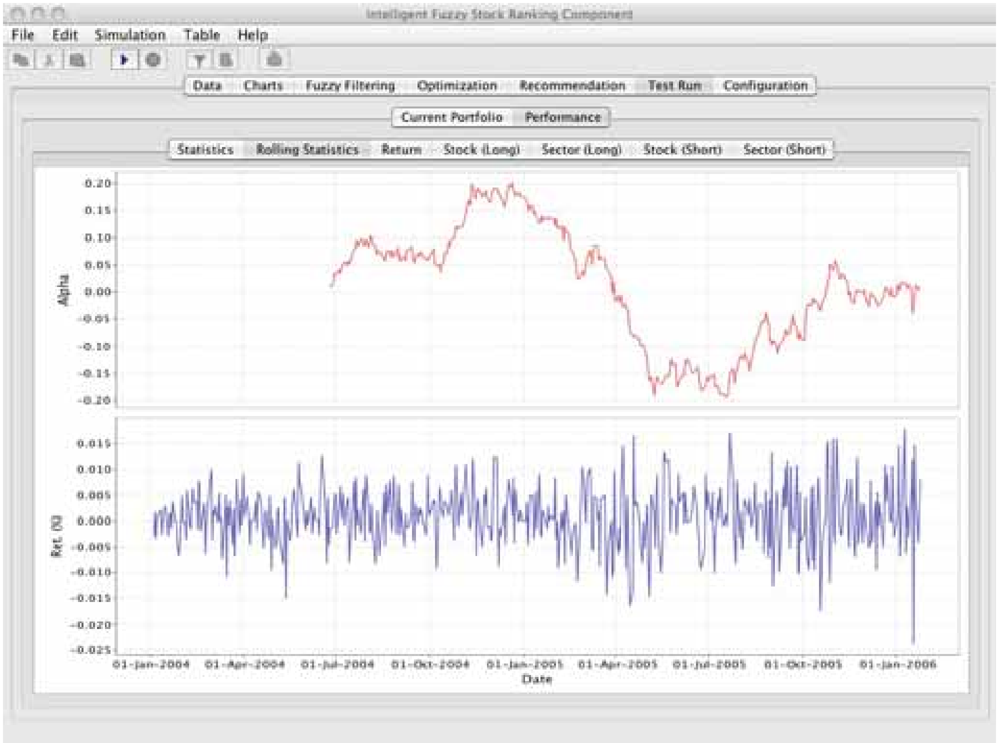


Figure 9.14: Performance (rolling) screen.

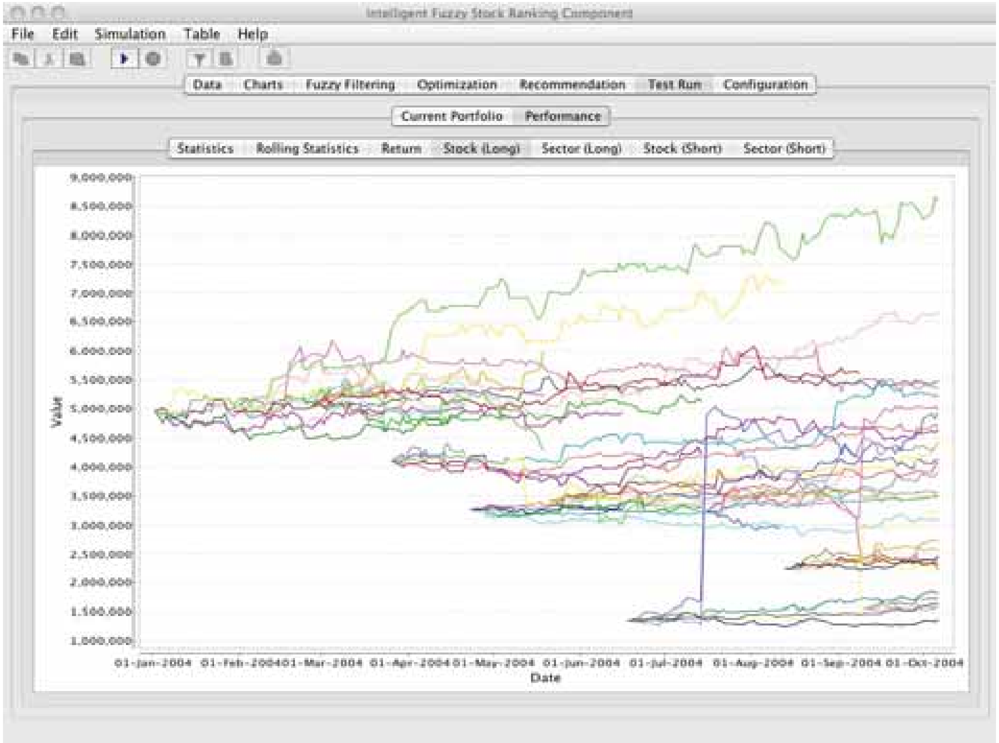


Figure 9.15: Long position value summary screen.

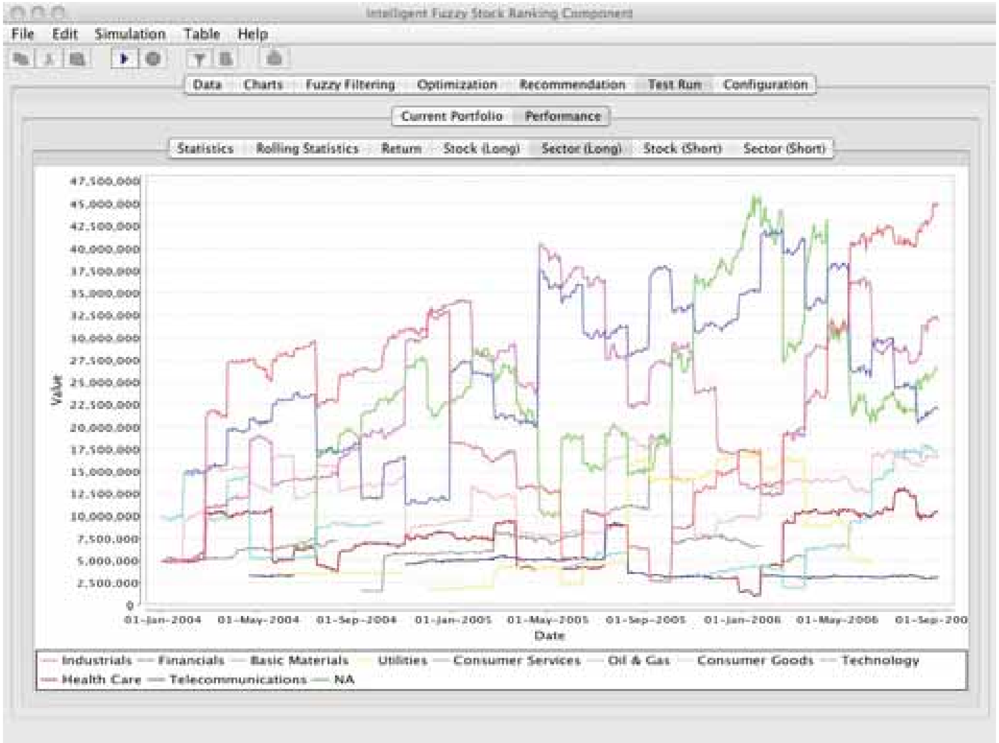


Figure 9.16: Long sector value screen.

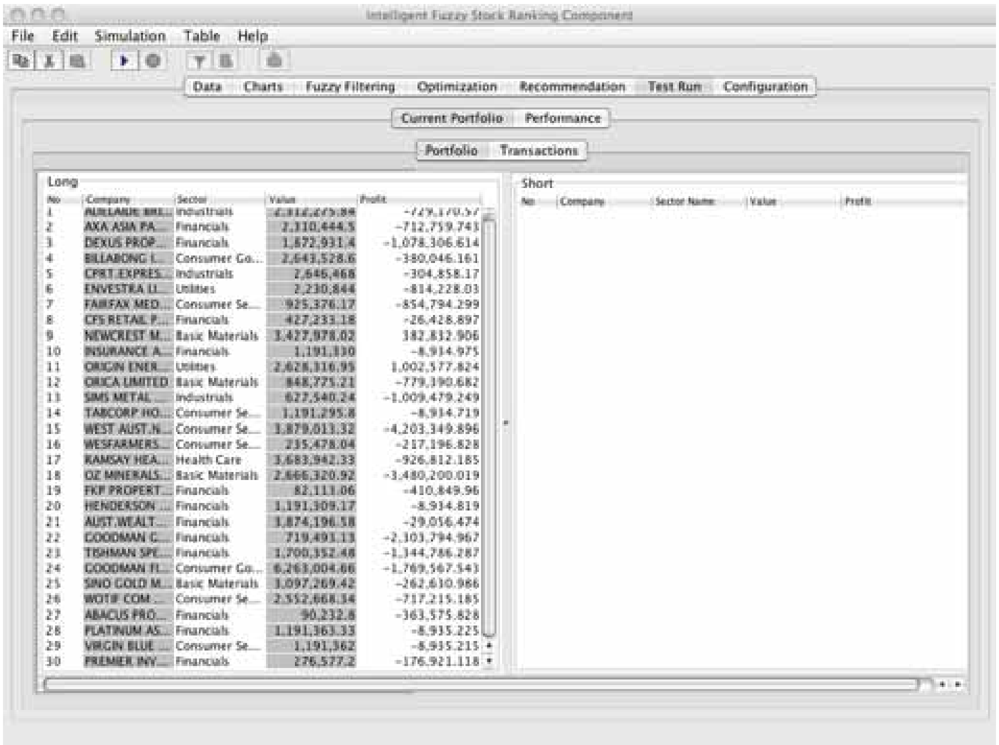


Figure 9.17: Portfolio contents tab.

9.2.7 System configuration

Properties for all aspects of the system are able to be controlled by configuration settings. These settings are divided into the following main configuration groups by functionality:

Input Data Files locations for input files for data base construction operation.

Rule Inputs/LVars Specification for controlling the size and contents of the information set (see Figure 9.18) — depth of the model.

Stocks to Use Enables selection of assets to use in the system individually or by industry group — breadth of the model.

Asset Filtering controls the application of user defined rule bases (see Figure 9.20).

Rule base Parameters to specify the basic properties for rule base strategies (either those evolved or user specified), the parameters are listed in Table 9.1.

Optimization configuration for the evolutionary search process, these parameters are provided in Table 9.2.

Historical Testing parameters specific to trading strategy implementations, these parameters are shown in Figure 9.21.

In the terminology used above *breadth* of a strategy model is defined as the set of assets used in its discovery; the *depth* refers to the number of explanatory variables considered.

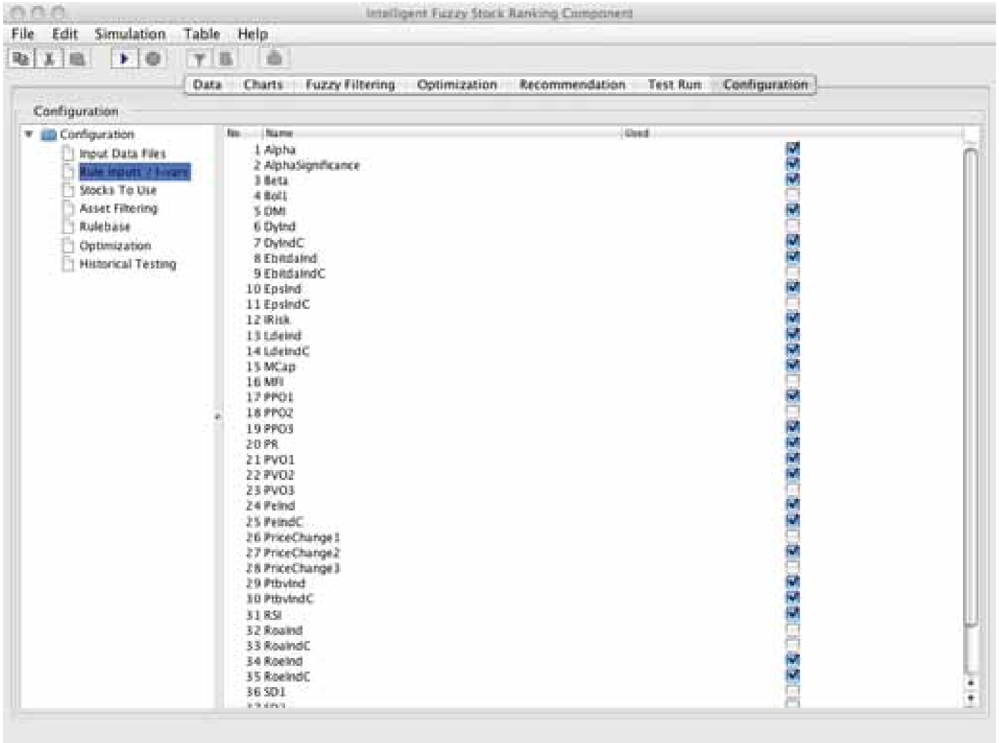


Figure 9.18: Input data and indicator selection screen.

Table 9.1: Parameters settable in the rule base configuration screen and meanings.

Parameter	Meaning
Max rules	maximum number of rules in a rule base
Min rules	minimum number of rules
Num output levels	output discretization
Max inputs	maximum number of lvar specifications in a single rule
Num MF	membership functions per variable
Adapt MF	Adapt MF for each variable in optimization

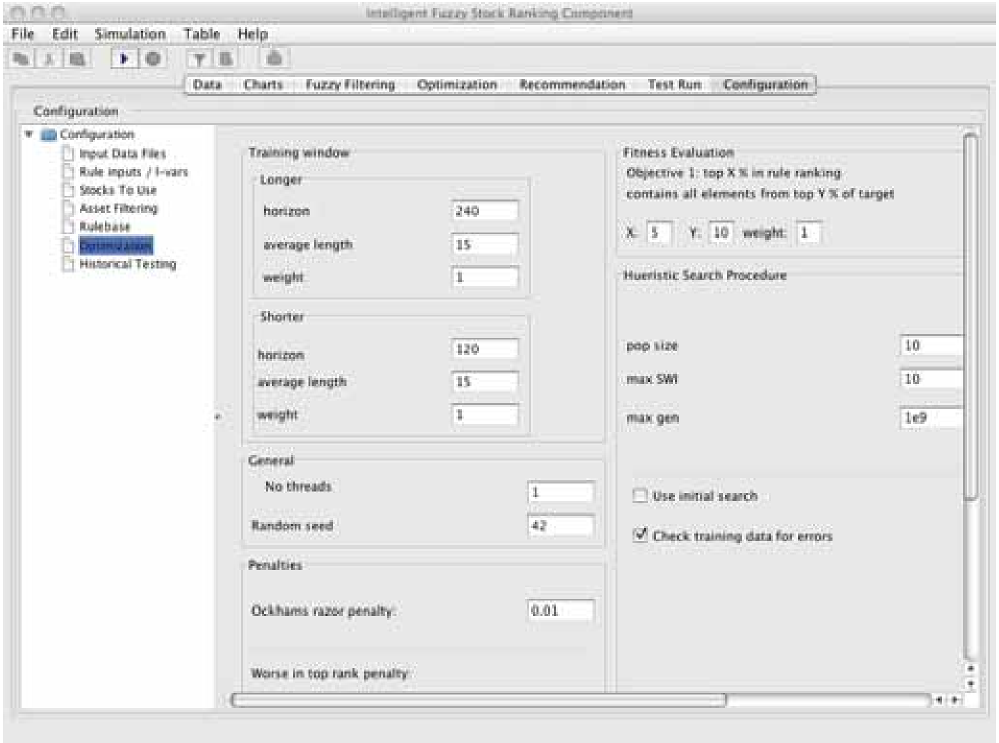


Figure 9.19: Optimization configuration screen.

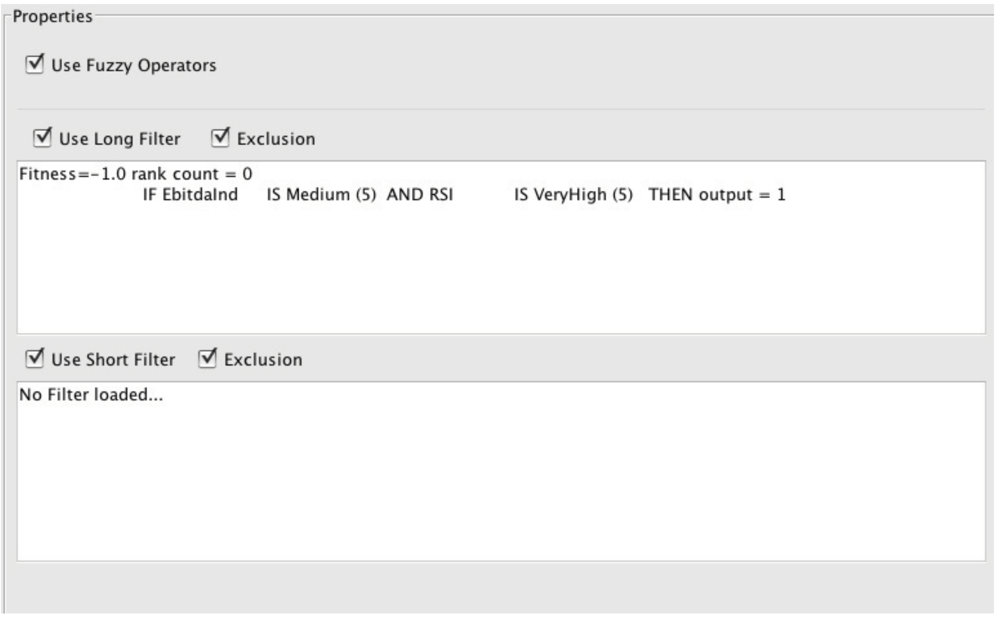


Figure 9.20: Combination of user and system rule bases. The setting to *use fuzzy operators* indicates whether the user defined rules exclude assets which do not match or if the user and system rule bases are combined. By setting the *exclusion* check box the user rule base acts to avoid choosing of matching assets.

Table 9.2: Parameters settable in the optimization configuration screen and meanings.

Parameter	Meaning
Window length	training window data size
Weight	weight in aggregation for solutions from this training window
Average length	number of testing days used from the training data
Random seed	seed for random number generator
No threads	number of threads to use by optimizer
Ockhams razor	size of the Ockhams razor penalty
Worse in top rank	penalty for strategies that select worse stocks
Evaluation function X	top X% of rule ranking
Evaluation function Y	target top Y% of ideal outcome in training
Population size	size of EA population
Max SWI	algorithm stopping condition steps without improvement
Max GEN	absolute max number of generations
Use initial search	seed the EA with the best single rule found by hill climber
HC after EA	a hill climber search to hone the solution after the EA run

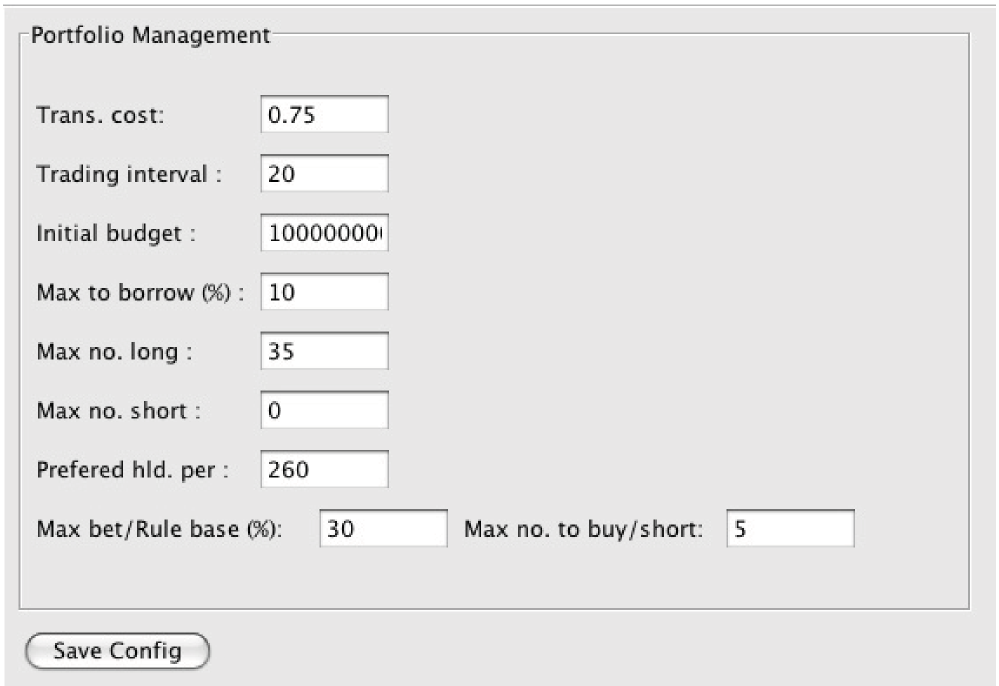


Figure 9.21: Portfolio management configuration. These settings control the application strategies for real trading and also in simulation. Some are self explanatory. The max long and short parameters constrain the maximum number of positions of these types to take in a portfolio. The preferred holding period defines the number of days positions are held in the absence of a conflicting signal. The max bet refers the percentage of available cash that can be "bet" on a strategy and the max to buy is the maximum number of positions to take based on a solution recommendation in each rebalancing event.

Chapter 10

Concluding Remarks and Future Work

In Chapters 2 and 3 we introduced and examined the portfolio management task. We showed that it is reasonable to anticipate the possibility of attaining useful fund management and asset allocation strategies that use computational intelligence. Academic financial literature (see the literature review in Chapter 3) contains evidence that strategies based on technical trading can outperform the market over certain periods and that this is most likely by exploiting the behavior of market participants. Therefore, it is not surprising that computational intelligence, and related methodologies, have been observed to learn successful trading strategies. We have also found that extending this approach to be adaptive is beneficial. These results are anticipated (and supported by) financial research such as, [16] which provides evidence of the intuition that it is an easier task to find rules that work over limited periods rather than rules that are able to perform at all times. A cornerstone of the approach we develop is the provision of a means to discover and exploit rules when they work and then discard them to be replaced by new working rules as time progresses.

The approach developed in this thesis makes use of mechanisms for updating portfolio management strategies as new data is fed in and with respect to performance of a managed portfolio over time. Because of this we propose a the paradigm of a quantitative investment forecasting model/strategy as *an information set and a heuristic search process together with system parameters* instead of the usual case involving a particular set of rules or mathematical formula. Furthermore, we also address another important issue in computational financial prediction modeling —

historical testing results for any type of heuristic system may be quite influenced by system settings. In the worst case this is a form of data snooping. This is clearly a significant issue in computational financial prediction as noted by research such as [105, 51, 50]. Mechanisms for self-adjusting parameters including solution properties (e.g. in the case of a fuzzy rule base the number of rules and fuzzy membership functions etc) reduce this problem by limiting the dependance of learned strategies on specific parameter settings. Essentially self adjusting/adaptive parameter setting is in this application class a method for increasing the generality of *an information set and a heuristic search process*.

Chapter 4 presented an evolving fuzzy system that enables a portfolio of selected stocks to be managed by computational intelligence. The approach involves an evolving Fuzzy System that conducts technical analysis of stocks based on price and volume data. An evaluation based on simulation was used for the evolutionary process that optimizes the rules. By applying an (evolved) fuzzy rule base to each stock in a market, on a particular day, the system ranks the stocks by output from the fuzzy rules. The stock ranking is interpreted to mean buy and sell recommendations. A portfolio is constructed and then re-balanced at set intervals such that the highest ranked stocks are bought and lower ranked stocks are sold. As stocks fall in ranking from the top buy recommendations, they are sold off and replaced with higher ranked stocks. In the prediction model optimization the fuzzy logic rule bases are evaluated using a fitness function that involves simulation of applying the rules in historic data. This function describes the ultimate objective of the portfolio. In this case, the fitness function is a return on investment measure. It is supplemented by a specific financial penalty function that penalizes solutions that select portfolios which experience significant losses in the interim of the testing period (the period during which the simulation evaluation takes is conducted) even if the final return is good. Effectively, it is a penalty for downside risk.

The empirical results from out-of-sample testing show that this approach can not only out perform traditional, fixed rule-based strategies such as price momentum and alpha based strategies but also the market index, see Chapter 2. This is shown for the case of MSCI Europe listed stocks spanning a period from 1990 until the end of 2005. Given that we impose both costs to trading and restrictions on how trades can occur, it is quite an impressive result.

In the introduction, we suggested the possibility that this research could extend and or enhance financial research. Using our methodology we were able to test the use of technical analysis trading in a novel way. In academic finance, research into the efficient markets hypothesis involving quantitative studies of trading strategies has principally been based on utilizing one specific type of rule in isolation to others. However, we notice that a cursory examination of any of the currently popular technical trading books or websites that investors read regularly cite that one should never place complete faith in a single trading rule, and instead check for "confirming" signals from other indicators. The problem from an academic point of view in finance is that it is considerably difficult to test the success of such a strategy where there are an endless supply of combinations of rules that could be put together to determine a functional trading strategy. Nevertheless, by-and-large, technical traders do exactly this, choosing a certain set of rules to determine trades. In this way the system presented in Chapter 4 has facilitated a quantitative analysis much more similar to the real world usage of technical trading than previous research (see Chapter 5).

Our hypothesis that technical trading rules can contribute to profitable outcomes when used in a changing manner which is also more akin to how actual practitioners trade with this type of information is supported by the results. Dynamic rules led to positive results over and above a static rule generation method and also rules that have been mentioned in the literature. Furthermore, we note that a hill climbing rule tuning heuristic did not perform well which supports the proposition that more advanced techniques applied in a cross disciplinary fashion can facilitate insight into financial research questions in a way not possible using simpler techniques and also lead to successful application designs.

We believe the success of the computational intelligent approach lies in its ability to adapt forecasting models to new market conditions. This is a significant advantage over fixed rule strategies because such a system can successfully pinpoint technical trading patterns that allow it to select stocks that are likely to outperform. In Chapter 6 the impact of adaptation is tested further. Some adaptive mechanisms are developed and the system is extended by a larger set of input data variables. Experimentation shows that an adaptive approach to portfolio management is able to outperform a non-adapting methodology signficantly.

In Chapter 7 we provided a description of a method for managing a long/short portfolio of assets suitable for investing in falling or rising markets and the results of tests in the Australian Stock Market. It was found that the system could perform better than the market index and random stock selection. In combining solutions to construct recommendation rankings the approach takes into account the performance of solutions in training (fitness), using a weighted sum. The method described also enables multiple objectives to be combined to construct portfolio decision recommendations (adjusted stock recommendation rankings).

In experiments we showed the approach with solution aggregation was able to perform similarly to the approach used earlier in the thesis for most of the test period. However, when the market fell rapidly during a global financial crisis (most clearly observable in 2008 and 2009) the portfolio managed using the single solution approach fell in value much more sharply than the multi-objective portfolio. This shows that by combining several solutions, including optimizations for minimal volatility, the system was able to construct a portfolio that was more robust to the market downturn.

Finally in Chapter 10, we introduce and describe an adaptive business intelligence system for portfolio management that uses various elements developed during this research. The system combines an adaptive framework with optimization and prediction. It implements the investment analysis rationale discussed in the task specification give early in the thesis (see Chapter 2, esp. Figure 2.2). A financial analyst can interact with the intelligent system through the user interface to produce evolving stock selection strategies and apply these for decision making. Important features that are implemented include providing the ability to combine user defined criteria with evolving strategies and also the ability to use the approach in different portfolio management styles (long, long/short and index tracking).

The thesis has contributed an approach to equity portfolio management that uses computational intelligence methodologies to implement financial reasoning, with a basis in financial research. A novel approach intimately linking financial and computing science methodologies facilitates a particular path to searching a space of semantically meaningful logical propositions tested and evaluated using simulation and methods close to the way solutions will be used. This cross disciplinary approach that combines computational intelligence and financial research to perform key procedures at the conceptual level (as opposed to actually executing trades, storing information, etc) in portfolio management.

In comparison with classical financial modeling approaches the approach enables adapting to changing environmental conditions and a non-linear solution representation. Compared with existing computational intelligence approaches the holistic approach combining financial research at all levels in the design promotes interpretable of fuzzy models that can be integrated with user insights and preferences. Adaptive mechanisms facilitate learning adaptive investment decision models that adapt to an interplay of market processes, application performance and the environment.

It is found that the methodology is able to provide outperformance over the market in the cases tested. All financial information systems are compared through comparison with the market benchmark. State-of-art systems used in investment banks an so forth are not available for comparison, however it is the case that results in many cases reported in this thesis involved out-performance over the market, including with risk adjusted measures, that are consistent with performance reported in the industry.

There are a number of avenues for future work to extend the research presented in this thesis. In the remainder of this Chapter we discuss several promising possibilities.

Portfolio management is essentially a multi-objective problem with two mostly conflicting objectives of minimizing risk while maximizing return. In addition other sub criteria used to measure these two objectives may also be considered as separate. A further set of categories of objectives could include social and environmental impact and other possibilities that could influence an investments suitability. This could be particularly useful in practice: for example, a fund manager may wish a portfolio to target specific metrics used to compare fund performance. We have found that aggregating solutions optimized (in training) to target various criteria is a useful method for effecting different forecasting objectives satisfactorily in Chapter 8. The implementation of multi-objective optimization algorithms that produce several solutions in a single run to construct aggregate models would enhance these methods in a number of ways. First of all, by reducing the time taken to produce recommendations; and, in addition, algorithms that produce a Pareto front to balance conflicting objectives could lead to interesting results.

Another very important area we identify for future research involves the development of methods for the application of the generated solutions in ways to maximize the potential of the approach. This includes methods for choice of possible asset decisions. This is because solutions do not necessarily perform equally well for prediction and sometimes fail and in addition may have quite different characteristics. We suggest the first problem could be solved by including probability analysis and sampling the rate of successful predictions (using different data from that used for generating the solution) in the interpretation and decoding of recommendations the approach could be enhanced. The problem of selecting assets with different characteristics could be approached using a fuzzy decision model that is designed to weigh different possibilities in the light of current positions and objectives. In addition, in order to reduce the risk associated with the approach a variety of methods for limiting the exposure of capital to any evolved single solution could be developed. One simple approach along these lines would be to include a parameter that limits the amount of cash that may be invested on the basis of any prediction model.

An alternative evaluation method that is quite promising could be based on probabilistic approaches to involve sampling periods of historic data to minimize prediction error rather than explicit simulation with a greater emphasis on periods with similar characteristics to very recent periods. In addition there are many other areas where alternative evaluation methods and other aspects of the approach could be further examined. Other promising extensions include the use of hierarchal fuzzy systems that impose additional structure on forecasting models, hybrid or extended solution representations that include more aspects of the task such as a flexible period over which recommendations hold (this would also have the effect of including an explicit sell signal as well as the buy signal in the recommendation) and the study and comparison of alternative algorithms and methodologies, especially in the area of multiple objectives.

List of Tables

List of Figures

Bibliography

[1] S. B. Achelis. *Technical Analysis from A to Z.* New York : McGraw Hill, c2001.

[2] F. Allen and R. Karjalainen. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, 51:245–271, 1999.

[3] A. Azzini and A. G. B. Tettamanzi. Evolving neural networks for static single-position automated trading. *J. Artif. Evol. App.*, 2008:1–17, 2008.

[4] T. Bäck. Adaptive business intelligence based on evolution strategies: some application examples of self-adaptive software. *Inf. Sci. Appl.*, 148(1-4):113–121, 2002.

[5] S. Baglioni, D. Sorbello, S. P. A.F. Capital, C. Da Costa Pereira, and A. G. B. Tettamanzi. Evolutionary multiperiod asset allocation. In *in Proc. Genetic and Evolutionary Computation Conf. (GECCO), 2000*, pages 597–604. Morgan Kaufmann, 2000.

[6] A. Bai. Determining fuzzy membership functions with tabu search – an application to control. *Fuzzy Sets and Systems*, 139(1):209–225, 2003.

[7] N. Barberis, A. Shleifer, and R. Vishny. A model of investor sentiment. *Journal of Financial Economics*, 49(3):307 – 343, 1998.

[8] G.F. Barrett and S.G. Donald. Consistent tests for stochastic dominance. *Econometrica*, 71(1):71–104, 2003.

[9] R.J. Bauer. *Genetic Algorithms and Investment Strategies.* John Wiley & Sons, Inc., New York, NY, USA, 1994.

[10] A. Bera and C. Jarque. Efficient tests for normality, heteroscedasticity, and serial independence of regression residuals. *Economics Letters*, 6:255–259, 1980.

[11] H. Bessembinder and K. Chan. Market efficiency and the returns to technical analysis. *Financial Management*, 27(2):5–17, 1998.

[12] L. Blume, D. Easley, and M. O'Hara. Market statistics and technical analysis: The role of volume. *The Journal of Finance*, 49(1):153–181, 1994.

[13] P.P. Bonissone, R. Subbu, and K.S. Aggour. Evolutionary optimization of fuzzy decision systems for automated insurance underwriting. In *Proc. of IEEE Int. Conf. on Fuzzy Systems FUZZ-IEEE02, 2002*, pages 1003–1008. IEEE, 2002.

[14] A. Brabazon and M. O'Neill. *Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series)*. pages 143–155,73–88, 2006.

[15] A. Brabazon and M. O'Neill. *Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series)*. pages 183–210, 2006.

[16] W. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47:1731–1764, 1992.

[17] S. J. Brown, W. N. Goetzmann, and A. Kumar. The Dow Theory: William Peter Hamilton's track record reconsidered. *The Journal of Finance*, 53(4):1311–1333, 1998.

[18] S.J. Brown, W.N. Goetzmann, and A. Kumar. The Dow Theory: William Peter Hamilton's track record re-considered. Yale school of management working papers, Yale School of Management, March 2004.

[19] M.G. Castillo Tapia and C.A.C. Coello. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 532–539, Sept. 2007.

[20] L. K. C. Chan, N. Jegadeesh, and J. Lakonishok. The profitability of momentum strategies. *Financial Analysts Journal*, 55(6):80–90, 1999.

[21] P Chang and C. Liu. A tsk type fuzzy rule based system for stock price prediction. *Expert Syst. Appl.*, 34(1):135–144, 2008.

[22] O. Cordón, F. A. C. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena. Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets and Systems*, 141(1):5–31, 2004.

[23] O. Cordón, F. Herrera, J. Montaña, A. M. Sánchez, and P. Villar. A prediction system for cardiovascularity diseases using genetic fuzzy rule-based systems. In *IBERAMIA*, pages 381–391, 2002.

[24] R. Davidson and J. Duclos. Statistical inference for stochastic dominance and for the measurement of poverty and inequality. *Econometrica*, 68(6):1435–1464, 2000.

[25] L. Davis. Handbook of genetic algorithms. 51, 1991.

[26] R. D'Mello, S.P. Ferris, and C.Y. Hwang. The tax-loss selling hypothesis, market liquidity, and price pressure around the turn-of-the-year. *Journal of Financial Markets*, 6(1):73–98, January 2003.

[27] Michayluk D.W. and R. Zurbruegg. Asymmetric volatility, correlation and returns dynamics between the US and UK securitised real estate markets. *Real Estate Economics*, 34(1), 2006.

[28] A.E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3:124–141, 1999.

[29] A.A.A Esmin and G. Lambert-Torres. Evolutionary computation based fuzzy membership functions optimization. In *Systems, Man and Cybernetics ISIC IEEE International Conference on*, pages 823–828. IEEE, 2007.

[30] E. F. Fama. Components of investment performance. *Journal of Finance*, 27(3):551–567, 1972.

[31] E.F. Fama and M.E. Blume. Filter rules and stock-market trading. *Journal of Business*, 1966.

[32] E.F. Fama and K. French. Common risk factors in the returns of stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

[33] E.F. Fama and K.R French. Multifactor explanations of asset pricing anomalies. *Journal of Finance*, 51(1):55–84, March 1996.

[34] C. Fyfe, J. Marney, and H. Tarbert. Technical analysis versus market efficiency a genetic programming approach. *Applied Financial Economics*, 9:183–191, 1999.

[35] R. Genay. Optimization of technical trading strategies and the profitability in security markets. *Economics Letters*, 59(2):249 – 254, 1998.

[36] A. Ghandar, Z. Michalewicz, M. Schmidt, T. To, and R. Zurbruegg. The performance of an adaptive portfolio management system. In *CEC 2008*. IEEE Press, 2008.

[37] A. Ghandar, Z. Michalewicz, M. Schmidt, T. To, and R. Zurbruegg. Computational intelligence for evolving trading rules. *IEEE Transactions On Evolutionary Computation*, 13(1), 2009.

[38] A. Ghandar, Z. Michalewicz, and R. Zurbruegg. Return performance volatility and adaptation in an automated technical analysis approach to portfolio management. *Int. J. Intell. Syst. Account. Financ. Manage.*, 16(1):127–146, 2009.

[39] J. Gomez. Evolution of fuzzy rule based classifiers. In *GECCO 2004*, 2004.

[40] T. H. Goodwin. The information ratio. *Financial Analysts Journal*, 54(4):34–43, 1998.

[41] R.C. Grinold and R.N.Kahn. *Active Portfolio Management*. McGraw-Hill: New York, 2 edition, 2000.

[42] N.R. Guo, T.S Li, and C. Kuo. Design of hierarchical fuzzy model for classification problem using gas. *Comput. Ind. Eng.*, 50(1):90–104, 2006.

[43] O. K. Gupta. Applications of quadratic programming. *Journal of Information and Optimization Sciences*, 16(1):177–194, 1995.

[44] R.G. Hagstrom. *The Warren Buffett way : investment strategies of the world's greatest investor*. J. Wiley, New York :, 1994.

[45] F. Herrera and E. Herrera-Viedma. Linguistic decision analysis: steps for solving decision problems under linguistic information. *Fuzzy Sets Syst.*, 115(1):67–82, 2000.

[46] F. Herrera and E. Herrera-Viedma. A takagi-sugeno fuzzy system for the prediction of river stage dynamics. *JARQ. Japan Agricultural Research Quarterly*, 40(4):369–378, 2006.

[47] M. Hollander and D.A. Wolfe. *Nonparametric Statistical Methods*. John Wiley and Sons, New York, 1999.

[48] L. Hsiang-Tai and J. Yoder. Optimal hedging with a regime-switching time-varying correlation garch model. *Journal of Futures Markets*, 27(5):495–516, 2007.

[49] C. Hsu, P.and Kuan. Reexamining the profitability of technical analysis with data snooping checks. *Journal of Financial Econometrics*, 3(4):606–628, 2005.

[50] P. Hsu and C. Kuan. Re-Examining the Profitability of Technical Analysis with White's Reality Check and Hansen's SPA Test. *SSRN eLibrary*, 2005.

[51] P. Hsu and C. Kuan. Reexamining the profitability of technical analysis with data snooping checks. *Journal of Financial Econometrics*, 3(4):606–628, 2005.

[52] H. Iba and N. Nickolaev. Genetic programming and polynomial models of financial time series. In *CEC 2000: Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, pages 1459–1466. IEEE Press, 2000.

[53] N. Jegadeesh and S. Titman. Profitability of Momentum Strategies: An Evaluation of Alternative Explanations. *Journal of Finance*, 56(2):699–720, 2001.

[54] R.I. John. Type 2 fuzzy sets: an appraisal of theory and applications. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(6):563 – 576, December 1998.

[55] R.W. Johnson, M.E. Melich, Z. Michalewicz, and M. Schmidt. Coevolutionary optimization of fuzzy logic intelligence for strategic decision support. *IEEE Transactions on Evolutionary Computation*, 9(6):682–694, 2005.

[56] B. A. Julstrom. Adaptive operator probabilities in a genetic algorithm that applies three operators. In *SAC '97: Proceedings of the 1997 ACM symposium on Applied computing*, pages 233–238, New York, NY, USA, 1997. ACM.

[57] B. Kamich. *How Technical Analysis Works*. New York: New York Institue of Finance, 2003.

[58] K.A. Kavajecz and E.R. Odders-White. Technical Analysis and Liquidity Provision. *Rev. Financ. Stud.*, 17(4):1043–1071, 2004.

[59] J Kroeske, A Ghandar, Z Michalewicz, and F Neumann. Learning fuzzy rules with evolutionary algorithms – an analytic approach. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 1051–1060, Berlin, Heidelberg, 2008. Springer-Verlag.

[60] R.K. Lai, C. Fan, W. Huang, and P. Chang. Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Syst. Appl.*, 36(2):3761–3773, 2009.

[61] J. Lakonishok, A. Shleifer, and R. W. Vishny. Contrarian investment, extrapolation, and risk. *Journal of Finance*, 49(5):1541–78, December 1994.

[62] F Larkin and C. Ryan. Good news: Using news feeds with genetic programming to predict stock prices. In *EuroGP*, pages 49–60, 2008.

[63] H. Leland. Beyond Mean-Variance: Performance Measurement in a Nonsymmetrical World. *Financial Analysts Journal*, pages 27–36, 1999.

[64] G. Lezos and M. Tull. Neural network and fuzzy logic techniques for time series forecasting. In *Computational Intelligence for Financial Engineering, 1999. (CIFEr) Proceedings of the IEEE/IAFE 1999 Conference on*, pages 191–197, 1999.

[65] M. Li, S. Azarm, and V. Aute. A multi-objective genetic algorithm for robust design optimization. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 771–778, Washington DC, USA., 2005.

[66] M. Lim and R.J. Coggins. Optimal trade execution: an evolutionary approach. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1045–1052 Vol. 2, Sept. 2005.

[67] A.W. Lo and A.C. MacKinlay. Data-Snooping Biases in Tests of Financial Asset Pricing Models. *The Review of Financial Studies*, 3(3):431–467, Fall 1990.

[68] A.W. Lo, H. Mamaysky, and J. Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*, 55(4):1705–1765, 2000.

[69] A.W. Lo, H. Mamaysky, and J. Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, 2000.

[70] B. C. Ludwig and K. Daehwan. *Quantitative Equity Portfolio Management*. McGraw Hill, 2006.

[71] Y. Lui and D. Mole. The Use of Fundamental and Technical Analyses by Foreign Exchange Dealers: Hong Kong Evidence. *Journal of International Money and Finance*, 17(3):535–545, June 1998.

[72] Dietmar G. Maringer. Constrained index tracking under loss aversion using differential evolution. In A. Brabazon and M. O'Neill, editors, *Natural Computing in Computational Finance*, volume 100 of *Studies in Computational Intelligence*, pages 7–24. Springer, 2008.

[73] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[74] H. Markowitz. Foundations of portfolio theory. *The Journal of Finance*, 46(2):469–477, 1991.

[75] B.R. Marshall, R.H. Cahan, and J.M. Cahan. Can commodity futures be profitably traded with quantitative market timing strategies? *Journal of Banking and Finance*, 32(9):1810 – 1819, 2008.

[76] Z. Michalewicz. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4:1–32, 1996.

[77] Z. Michalewicz. Parameter control in practice. *Springer Series Studies in Computational Intelligence*, 2007.

[78] Z. Michalewicz, M. Schmidt, M. Michalewicz, and C. Chiriac. *Adaptive Business Intelligence*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[79] Z. Michalewicz, M. Schmidt, M. Michalewicz, and C. Chiriac. Adaptive business intelligence: three case studies. In Y.S. Ong S. Yang and Y. Jin, editors, *Evolutionary Computation in Dynamic and Uncertain Environments*, Studies in Computational Intelligence. Springer, 2007.

[80] Z. Michalewicz, M. Schmidt, M.and Michalewicz, and C. Chiriac. A decision-support system based on computational intelligence: A case study. *IEEE Intelligent Systems*, 20(4):44–49, 2005.

[81] Z. Michalewicz, M. Schmidt, M.T. Michalewicz, and C. Chiriac. Case study: An intelligent decision-support system. *IEEE Intelligent Systems*, 20(4):44–49, 2005.

[82] M. Mucientes, L. Moreno, A. Bugar, and S. Barro. Evolutionary learning of a fuzzy controller for wall-following behavior in mobile robotics. *Soft Comput.*, 10(10):881–889, 2006.

[83] S.K. Nanda, D.P. Tripathy, and S. K. Patra. Fuzzy inference system-based noise prediction models for opencast mines. *International Journal of Mining, Reclamation and Environment*, 6(6):1748–0930, December 2009.

[84] C. Neely, P. Weller, and R. Dittmar. Is technical analysis in the foreign exchange market profitable? a genetic programming approach. *The Journal of Financial and Quantitative Analysis*, 32(4):405–426, 1997.

[85] C. J. Neely. Risk-adjusted, ex ante, optimal technical trading rules in equity markets. *International Review of Economics and Finance*, 12(1):69–87, 2003.

[86] C.J. Neely and P.A. Weller. Predicting exchange rate volatility: genetic programming vs. garch and riskmetrics. Technical report, 2001.

[87] C.J. Neely and P.A. Weller. Using a genetic program to predict exchange rate volatility. In Shu-Heng Chen, editor, *Genetic Algorithms and Genetic Programming in Computational Finance*, chapter 13, pages 263–279. Kluwer Academic Publishers, Dordrecht, July 2002.

[88] M. O'Neil, A. Brabazon, C. Ryan, and J. Collins. Evolving market index trading rules. *Lecture Notes in Computer Science: Applications of Evolutionary Computing*, 5(4):343–353, 2001.

[89] M. O'Neill and M. Cleary. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358, 2001.

[90] C. Park and Scott H.I. A reality check on technical trading rule profits in us futures markets. Technical report, 2005.

[91] W. Pedrycz. Why triangular membership functions? *Fuzzy Sets Syst.*, 64(1):21–30, 1994.

[92] L. Piotr. Evolutionary strategies for building risk-optimal portfolios. In *Natural Computing in Computational Finance*, pages 53–65. 2008.

[93] D. Quintana, C. Luque, and P. Isasi. Evolutionary rule-based system for ipo underpricing prediction. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 983–989, New York, NY, USA, 2005. ACM.

[94] D. Rafal and S. Leszek. Agent-based co-operative co-evolutionary algorithm for multi-objective optimization. In *ICAISC*, pages 388–397, 2008.

[95] M. Ratner and R.P.C. Leal. Tests of technical trading strategies in the emerging equity markets of latin america and asia. *Journal of Banking and Finance*, 23(12):1887–1905, December 1999.

[96] E. Ruspini, P.P Bonissone, and W. Pedrycz. Handbook of fuzzy computation. 51, 1998.

[97] M.F. Seyyed and Z.J. Mansoor. Constructing accurate fuzzy classification systems: A new approach using weighted fuzzy rules. In *Computer Graphics, Imaging and Visualization, CGIV 2007*, pages 408–413, 2007.

[98] W.F. Sharpe. Mutual fund performance. *Journal of Business*, 39(1):119–138, 1966.

[99] Y. Shi, R. Eberhart, and Y. Chen. Implementation of evolutionary fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 7(2):109–119, 1999.

[100] J.D. Stone, T.R. Robinson, J.E. Pinto, and D.W. McLeavey. *Equity Asset Valuation*. John Wiley and Sons, Inc, 2002.

[101] R. Sullivan, A. Timmermann, and H. White. Data-snooping, technical trading rule performance, and the bootstrap. *Journal of Finance*, 54:1647–1691, 1997.

[102] Takagi T. and Sugeno M. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1):116–132, January/February 1985.

[103] J.D. Thomas and K. Sycara. GP and the predictive power of internet message traffic. In Shu-Heng Chen, editor, *Genetic Algorithms and Genetic Programming in Computational Finance*, chapter 4, pages 81–102. Kluwer Academic Press, 2002.

[104] J.R. Trigueros. Extracting earnings information from financial statements via genetic algorithms. In *Computational Intelligence for Financial Engineering, 1999. (CIFEr) Proceedings of the IEEE/IAFE 1999 Conference on*, pages 281–296, 1999.

[105] H. White. A reality check for data snooping. *Econometrica*, 68(5):1097–1126, 2000.

[106] J.W. Wilder. *New Concepts in Technical Trading Systems*. Trend Research, 1978.

[107] C. Wu and G. Liu. A genetic approach for simultaneous design of membership functions and fuzzy control rules. *J. Intell. Robotics Syst.*, 28(3):195–211, 2000.

[108] R. Yager. A general approach to rule aggregation in fuzzy logic control. *Appl. Intell.*, 2(4):333–351, 1992.

[109] W. Yan and C.D. Clack. Evolving robust gp solutions for hedge fund stock selection in emerging markets. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 2234–2241, New York, NY, USA, 2007. ACM.

[110] J. Yen, L. Wang, and C. W. Gillespie. Improving the interpretability of TSK fuzzy models by combining global learning and global learning. 6(4):530–537, 1998.

[111] M. H. Zarandi F., B. Rezaee, I. B. Turksen, and E. Neshat. A type-2 fuzzy rule-based expert system model for stock price analysis. *Expert Syst. Appl.*, 36(1):139–154, 2009.

[112] J. Zhang and R.A. Stine. Autocovariance structure of markov regime switching models and model selection, 1999.