Investigations into Novel Strategies for Intelligent Agent-based Decision Making

Cyril Schoreels, BSc. (Hons.)

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

June 2008
Abstract

The role of automation in modern trading environments has increased dramatically over the last decade, even exhibiting a fundamental shift in market behaviour as a direct consequence of algorithmic trading and a significant increase in trading frequency. As a self-propagating consequence, the need for automation in both analysis and reactivity to market movements is one of the fastest developing areas in industry and the field of computer science.

A large variety of approaches exist in analysing market data and deriving indications as to the market’s future movement. The focus of this work addressed the choice of trading methodology in simulation, using popular technical analysis tools, the capital asset pricing model and a hybrid implementation of the two. The algorithmic design of the genetic algorithm used in evolving agent populations and the choice and impact of the associated fitness function was studied in depth. Furthermore, the choice of a static approach, in which agents are first trained and then exposed to testing data, versus that of an adaptive approach, in which agents are continuously retrained, was analysed. Lastly, the impact of amalgamating trading decisions made by individual trading agents was studied, using a variety of simple voting methods compared to the performance of those agents individually.

Findings clearly demonstrated that the hybrid trading methodology showed significant benefits over its constituents, while noticeable differences were observed in agent performance as a result of algorithm design and choice of fitness functions. Furthermore, the adaptive system showed itself clearly superior to the static. Lastly, the performance of derived amalgamated decisions proved inferior to that of the individual agents.
Acknowledgements

I would like to thank my supervisor, Dr. Jon Garibaldi, for some excellent advice and guidance on my work and the research process. I would also like to thank the University of Nottingham, for providing the funding and facilities to conduct my research as well as an exceptional study environment.
Contents

List of Tables v

List of Figures vii

1 Introduction 1
  1.1 Introduction and Background to Research 1
  1.2 Aims of this Work 5
  1.3 Structure of the Thesis 11
  1.4 Definitions of Key Concepts 12
  1.5 Summary 13

2 Literature Review 14
  2.1 Introduction 14
  2.2 Agent Technology 16
  2.3 Artificial Intelligence 20
  2.4 Intelligent Agents in Financial Markets 28
  2.5 Decision Making Theorems 62
  2.6 Summary 67

3 Experimentation System Outline 69
  3.1 Introduction 69
  3.2 Agent Design 70
  3.3 Historical Market Data 71
  3.4 Evolutionary Approaches 72
  3.5 Fitness Measures 80
  3.6 Methodologies 85
  3.7 Statistical analysis 104
  3.8 Summary 106

4 An Investigation into the Use of Different Trading Methodologies 107
  4.1 Introduction 107
CONTENTS

4.2 System Outline 108
4.3 Method and Results 108
4.4 Analysis and Discussion 114
4.5 Summary and Conclusions 117

5 An Investigation into the Impact of Genetic Algorithms 118
  5.1 Introduction 118
  5.2 Genetic Algorithms Investigated 119
  5.3 Fitness Types Investigated 140
  5.4 Summary and Conclusions 147

6 An Investigation into Static and Adaptive Systems 149
  6.1 Introduction 149
  6.2 Static System 150
  6.3 Adaptive System 166
  6.4 Comparison of Static and Adaptive Agent Systems 172
  6.5 Conclusion 184
  6.6 Summary 186

7 An Investigation into Centralized Decision Making 188
  7.1 Introduction 188
  7.2 Central decision maker agent 189
  7.3 Experimentation and results 190
  7.4 Analysis and discussion 195
  7.5 Summary and conclusions 196

8 Conclusions 198
  8.1 Contributions 198
  8.2 Limitations and Further Work 202
  8.3 Reflections on the Original Hypotheses 205
  8.4 Final Remarks 209
  8.5 Dissemination 209

References 211

A Glossary of Terms 234

B Training Genome Convergence 239
  B.1 Genome Convergence using Technical Analysis 239
  B.2 Genome Convergence using Hybrid 246
  B.3 Genome Convergence using the CAPM 253
  B.4 Genome Convergence using Sharpe 260
List of Tables

3.1 Securities used in the system ............................................. 72
3.2 Data splits used in the system ........................................... 73
3.3 Defines the cardinality, length and function of each gene .......... 87
3.4 Defines the cardinality, length and function of each gene .......... 99
3.5 Defines the cardinality, length and function of each gene .......... 101
3.6 Defines the cardinality, length and function of each gene .......... 104

4.1 Capital Asset Pricing Model Results ..................................... 110
4.2 Technical Analysis Results ................................................ 111
4.3 Capital Asset Pricing Model and Technical Analysis hybrid results . 112
4.4 Results Summary ............................................................ 113
4.5 Mann-Whitney U tests ...................................................... 114

5.1 Results comparing agent performance by genetic algorithm used ... 121
5.2 Run-time observations for each genetic algorithm .................... 133
5.3 Statistical analysis of algorithms across methodologies .......... 134
5.4 Rank based analysis of algorithms across methodologies .......... 134
5.5 Rank based analysis of algorithms in training across methodologies . 137
5.6 Mean performance by fitness type and algorithm .................... 142
5.7 Mean fitness type performance ........................................... 143
5.8 Mean fitness type performance ........................................... 144
5.9 Rank based analysis of results .......................................... 145

6.1 Data used for training and testing the agent population .......... 152
6.2 Experiment description ...................................................... 152
6.3 Results ................................................................. 154
6.4 Mann-Whitney U Test P values ......................................... 155
6.5 Results for fixed agents .................................................. 159
6.6 Statistical analysis of evolved versus fixed agents ................. 160
6.7 Statistical analysis of evolved versus fixed agents without Training A results .................................................. 160
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.8</td>
<td>Parameters used and corresponding results</td>
<td>163</td>
</tr>
<tr>
<td>6.9</td>
<td>Comparison by data set</td>
<td>163</td>
</tr>
<tr>
<td>6.10</td>
<td>Comparison by elite</td>
<td>163</td>
</tr>
<tr>
<td>6.11</td>
<td>Comparison by removed</td>
<td>164</td>
</tr>
<tr>
<td>6.12</td>
<td>Parameters used and corresponding results</td>
<td>164</td>
</tr>
<tr>
<td>6.13</td>
<td>Comparing immigration versus mutation</td>
<td>165</td>
</tr>
<tr>
<td>6.14</td>
<td>Variations in mutation rate</td>
<td>165</td>
</tr>
<tr>
<td>6.15</td>
<td>Consistency across data sets</td>
<td>165</td>
</tr>
<tr>
<td>6.16</td>
<td>Experiment’s parameters and results</td>
<td>170</td>
</tr>
<tr>
<td>6.17</td>
<td>Kruskal-Wallis Test Results</td>
<td>171</td>
</tr>
<tr>
<td>6.18</td>
<td>Static and adaptive system results</td>
<td>177</td>
</tr>
<tr>
<td>6.19</td>
<td>Summary of results</td>
<td>177</td>
</tr>
<tr>
<td>6.20</td>
<td>Trading activity</td>
<td>180</td>
</tr>
<tr>
<td>6.21</td>
<td>Mann-Whitney U Test Results</td>
<td>181</td>
</tr>
<tr>
<td>6.22</td>
<td>Agent performance versus benchmarks</td>
<td>183</td>
</tr>
<tr>
<td>7.1</td>
<td>Experiment results (all values x10^8)</td>
<td>192</td>
</tr>
<tr>
<td>7.2</td>
<td>Kruskal-Wallis test</td>
<td>195</td>
</tr>
<tr>
<td>7.3</td>
<td>Mann-Whitney U tests</td>
<td>195</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Examples of crossover and mutation in genetic algorithms . . . . . . . 22
2.2 Impact on total fitness by introducing age or finite lifespans to solutions 26
2.3 An illustrative schematic outline of an Artificial Neural Network . . . 33
2.4 Simple function represented in a tree structure . . . . . . . . . . . . . 48
3.1 Data splits used . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
3.2 Original Genetic Algorithm . . . . . . . . . . . . . . . . . . . . . . 76
3.3 Decision process from technical analysis to acquisition and sale of securities . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97
5.1 Genetic algorithms performance by generation using technical indicators . 123
5.2 Genetic algorithms performance by generation using the hybrid approach . 124
5.3 Genetic algorithms performance by generation using the CAPM . . . . 125
5.4 Genetic algorithms performance by generation using the Sharpe ratio . 126
5.5 Technical Indicators Genome Correlation Overview . . . . . . . . . . 128
5.6 Hybrid Genome Correlation Overview . . . . . . . . . . . . . . . . 129
5.7 CAPM Genome Correlation Overview . . . . . . . . . . . . . . . . 130
5.8 Sharpe Genome Correlation Overview . . . . . . . . . . . . . . . . 131
6.1 Diminishing returns of training . . . . . . . . . . . . . . . . . . . . . 151
6.2 Adaptive system . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
6.3 Average performance graphs . . . . . . . . . . . . . . . . . . . . . . 179
7.1 Decision process overview . . . . . . . . . . . . . . . . . . . . . . . 191
7.2 Average performance of CDMs . . . . . . . . . . . . . . . . . . . . . 194
B.1 TI oGA Genome Correlation . . . . . . . . . . . . . . . . . . . . . . 240
B.2 TI sGA Genome Correlation . . . . . . . . . . . . . . . . . . . . . . 241
B.3 TI tGA Genome Correlation . . . . . . . . . . . . . . . . . . . . . . 242
B.4 TI rsGA Genome Correlation . . . . . . . . . . . . . . . . . . . . . . 243
B.5 TI rsGAs Genome Correlation . . . . . . . . . . . . . . . . . . . . . 244
B.6 TI rtGAs Genome Correlation . . . . . . . . . . . . . . . . . . . . . 245
LIST OF FIGURES

B.7 Hybrid oGA Genome Correlation ........................................ 247
B.8 Hybrid sGA Genome Correlation ........................................ 248
B.9 Hybrid tGA Genome Correlation ........................................ 249
B.10 Hybrid rsGA Genome Correlation ...................................... 250
B.11 Hybrid rsGAfs Genome Correlation .................................... 251
B.12 Hybrid rtGAfs Genome Correlation .................................... 252
B.13 CAPM oGA Genome Correlation ....................................... 254
B.14 CAPM sGA Genome Correlation ....................................... 255
B.15 CAPM tGA Genome Correlation ....................................... 256
B.16 CAPM rsGA Genome Correlation ....................................... 257
B.17 CAPM rsGAfs Genome Correlation ..................................... 258
B.18 CAPM rtGAfs Genome Correlation ..................................... 259
B.19 Sharpe oGA Genome Correlation ...................................... 261
B.20 Sharpe sGA Genome Correlation ...................................... 262
B.21 Sharpe tGA Genome Correlation ...................................... 263
B.22 Sharpe rsGA Genome Correlation ...................................... 264
B.23 Sharpe rsGAfs Genome Correlation .................................... 265
B.24 Sharpe rtGAfs Genome Correlation .................................... 266
CHAPTER 1

Introduction

1.1 Introduction and Background to Research

Decision support and the related more fully automated trading strategies are relatively new concepts in the long history of trade. However, they have established themselves universally as trading aids in industry to the extent of being a necessity in remaining a competitive force for most market participants. With ever increasing trading speeds and ever increasing competition, heuristics and agent based principles have been permeating this area and are argued to form the next generation of automated trading strategies and systems. Forming part of the field of computational finance in general, a lot of effort thus far has been focused on the ability to rapidly execute trades and process significant volumes of data to support decision-making as a crucial element in maintaining a competitive edge and maximizing profit margins. As markets represent either physical or virtual meeting places for buyers and sellers
to interact and exchange assets of various forms, it is becoming increasingly attractive for the long-term to not only use computer simulations to study market characteristics, but also to develop systems that interact both directly and indirectly with human counterparts. As for instance with the advance of straight-through-processing becoming more common place [102], a human’s role in basic data analysis has continued to become less prominent.

Another issue of interest is the complexity of modeling socio-economic environments, which raises further questions on how to best design systems for machine simulations. Research has explored a multitude of approaches emulating environments such as whole economies or more specific areas such as equity markets, hoping to create better understanding of how they operate and what the main factors are that determine their behavior. These underlying factors are of key interest to governments and corporations. This work attempts to further this understanding by investigating the predictive capabilities of simulated traders.

Multi-agent technologies have lent themselves well and are naturally suited to duplicating market environments, equipped with various levels of autonomy as well as being able to simulate interactions between people and the environment well. A vast array of literature has explored the application and use of agents in many fields, with an overview presented in [145]. Agents have frequently been coupled with artificial intelligence methods [38], in particular genetic algorithms [12], to evolve solutions to problems.
Overall, there exist a multitude of different approaches to automated trading, employing artificial neural networks, learning classifier systems, case-based reasoning, genetic algorithms and others [97, 109, 88]. In financial markets, the tendency has been to employ neural networks, motivated by being able to treat them as a “black box” and keeping the internal workings hidden and not having to conceptualize rules, such as in [129, 52]. A good general overview of research in the field was provided by Wong and Selvi [142]. Genetic algorithms in comparison are well equipped to deal with irregularities and exceptions in data, and given that markets are affected by expectations and at times irrational human behavior, this more robust option would seem more appropriate in this environment. Consequently, the use of genetic algorithms has been one of the most prolific in literature.

Similar work to the system used here was first published by Allen and Karjalainen [4], where they also used a genetic algorithm to evolve a set of trading strategies. They found that when transaction costs are included, their system did not earn excess returns compared to a buy-and-hold strategy. The buy-and-hold strategy is a common benchmark for performance comparison as it represents an investment strategy where the entire starting capital is invested in its entirety equally into all securities, without making any further trading decisions thereafter. Other authors in later work however demonstrated that their proposed approaches did manage to offer a viable alternative to buy-and-hold [13, 63, 98]. The research presented here differs by evolving actual traders rather than strategies, focusing on the internal structure
of these agents and the impact of the systems parameters on their performance.

With a wealth of literature discussing these issues in finance and related fields, this diversity is reflected in system implementations used for market studies, portfolio optimization problems and other computer aided investigations. However few studies exist determining the impact of the choices made in determining systems' components, and equally few seem to cite any prior work in justification for their chosen approach. The work presented here attempts to address these issues as well as provide an indication to the importance of these design choices in addressing the wide variety of problems tackled by contemporary automated systems.

To provide a general overview of other related work, a wide range of research exists on price prediction as well as volatility and risk analysis [33, 53, 56, 75]. This also includes behavioral analysis of markets, aiming to provide understanding of how a particular market works. Further research has been done on optimization problems aiming to deduce an optimal trading pattern to create successful portfolios. One of the most well known projects in artificial markets is the Santa Fe Stock Market, in which simple market mechanics allowing agents to make trading decisions using classifier systems are studied [9]. By generating demand functions for shares and using other forecasting methods, agents there are able to trade within the market, creating similar patterns to those observed in real markets. However, due to the complexity of the artificial market, the actual causes are hard to determine and further increase the difficulty in relating this back to real markets [69].
There have been many other projects looking at simulating traders and strategies in virtual markets using historical market data [111, 77, 64, 134]. A greater focus however is placed on the comparison of approaches and strategies in the Penn-Lehman Automated Trading Project [61]. The project offers a platform for different strategies to compete on equal terms and establish a comparative measure of success among them. Initial work has shown promising results with a majority of entered strategies showing positive earnings. Other projects have focused on different aspects of artificial markets, ranging from studying agent behavior within markets [108], to analysing market behavior and interactions using agents trading across national boundaries [7].

1.2 Aims of this Work

Though initially the aim was to explore the decision making behavior of agents in individualistic and group environments, a variety of studies were performed. These investigated a comparison of genetic algorithm models and the choice of fitness function, the performance of agents using static and adaptive approaches, ceteris paribus, as well as the performance of centralized decision makers drawing on groups of homogeneous agents.

A key novelty introduced in this work was the system implementation, combining a variety of trading methodologies and variations of genetic algorithms with otherwise homogeneous agents emulating real life traders. Compared to other work introduced later, it formed a non-strategy orientated evolutionary system with its focus instead
on the evolution of agents and their trading behavior. This was achieved through variations in the agents’ approach to security analysis and decision making behavior.

Based on this approach to portfolio optimization, a series of investigations then aimed at clearly establishing whether any differences could be observed in the system when employing variations on its implementation. The first investigation discussed here assessed the impact the choice of trading methodology might have on results. Its aim being to determine whether the choice of technical analysis, capital asset pricing model or a hybrid of the two approaches would actually affect performance. In wider literature justification of methodology is often based on its general use in industry or simply due to being a common approach. Few studies clearly attempt to establish whether any of the choices actually provide a clear advantage over the other in a given implementation with all other factors being left unchanged.

Similarly, the second investigation focused on determining whether the choice of genetic algorithm, and in a further part the choice of fitness function, would impact on performance. Though this could be expanded to include a variety of different factors broadening the investigation, the focus here was solely on variations of perhaps the most common form of genetic algorithm, as genetic algorithms lend themselves well to agent based systems. Often this is another neglected area of investigation in related research despite the potential improvement in exploring the search space and therefore solution quality.

Another important factor in the analysis of portfolio optimization simulations is
the assessment of performance. A variety of fundamentally different measures can be taken which will naturally impact, to a greater or lesser degree, on the assessment of an agent’s trading success or failure. Though it is very common and perhaps initially more intuitive to use a simple profit function, there are fundamentally different measures which can be argued to be equally or perhaps more reflective of an agent’s performance. Such as the area under a total assets’ graph or the ratio of relative changes in total assets on a day to day basis, as discussed later in Chapter 3.5. In light of this, a sample set of functions are used to assess an agents success in trading and implemented as their fitness measure in the genetic algorithms. A comparison is made as to which fitness type, if any, evolves the most successful agent population. This provides an elementary key to increasing the success of trading agents, fairly independently of the system implementation actually used. Particularly in light of trading competitions becoming increasingly popular, surprisingly little work has been done in this area.

One of the primary aims of this work was the comparison of using static versus adaptive systems. A static system representing a system evolved on a set of training data and then exposed to a set of testing data, while an adaptive system was based on an agent population being evolved on a shorter set of training data but being constantly retrained after every trading day. A plethora of systems exist in related work that are based on either approach with both groups presenting well balanced arguments in favour of their chosen approach. Though the static system appears to
be more common, in this author’s opinion many arguments in favour of an adaptive
system have been presented, with the “EUropean Network on Intelligent TExhnologies
for Smart Adaptive Systems” (EUNITE) project in particular having provided an
abundant source of arguments in its favour. The EUNITE project, which concluded
in June 2004, was a foray into increasing the understanding of adaptive and hybrid
systems and their potential applications [71]. One of its main tasks was to provide
direction and open up new areas of research in the domain, as well as provide access
to non-experts in applying intelligent technologies to problems. A significant amount
of research was published within its context. Many studies have shown that adaptive
systems can pose significant advantages to static systems in certain environments,
yet no study has performed a thorough comparison using otherwise identical systems
in the portfolio optimization domain. This work hopes to provide an insight into
whether either approach may offer benefits when used in a trading simulation and
discuss their merits under varying conditions.

Lastly, a brief exploration of centralizing trading decisions into a single opera-
tive unit is performed, based on the recommendations of a trading agent population.
Drawing an analogy to decision making among humans, benefits can be gained by
assembling a team of experts to the extent that this team’s joint decisions will outper-
form any individual team member. This investigation forms a brief foray into whether
these benefits can also be observed in an artificial trader environment or whether it
will simply represent an over abstraction of what is essentially an agent’s egocentric
decision behavior. In wider agent based work it is common to create centralized decision makers that take on specific roles in a multi agent system, however their use is not universally beneficial or detrimental to the overall system.

1.2.1 Conjectures and Hypotheses

As mentioned, in wider literature the choice of system design and its components often lacks comparative investigations into arguably equally suitable alternatives. In a large part, one of the key contributions of this thesis is to highlight the importance of these choices, as well as extensively exploring and thoroughly documenting the impact of such choices. The areas investigated represent some of the more frequently used concepts in research, which are directly studied in order to explore the changes in results as a direct consequence of these design choices. The related research is presented and discussed in Chapter 2.

The conjectures broadly addressed in this work can be summarized as follows, with a more descriptive summary below:

- The choice of trading methodology will affect agent performance, with one method likely comparatively superior.

- Small variations in the design of the genetic algorithm, as well as its parameters, can have a significant impact on agent performance in a system.

- The choice of fitness function and particularly when used simultaneously as the
measure of performance, has a significant impact on results and consequently derived conclusions.

- An adaptive system offers greater advantages over its static equivalent and will demonstrate better performance.

- Amalgamating multiple agents’ trading decisions into one, by means such as consensus voting for example, will merely reduce trading volume and not improve overall results.

In Chapter 4, it is implied that the use of different trading methodologies will affect the performance of an agent. The conjecture here is that one approach used should distinguish itself as superior, or to some degree, approaches might present characteristics that would allow a more effective use based on the application environment.

In Chapter 5, the evolutionary system used is scrutinized as to whether changes in the evolutionary system would display characteristics that might offer advantages compared to others. Taken in two parts, in the first it is theorized that modification of the genetic algorithm implementation may provide equally competitive results, or better, while reducing the resources required. This would be of particular use in time critical or resource constraint applications such as in trading competitions. In the second part it is conjectured that the choice of fitness function has an immediate effect on the solutions found and therefore the performance observed. It is hoped that
1. introduction

the less conventional and common measures may suggest equally or more desirable
alternatives to the most popular assessment usually used.

In Chapter 6, a static versus an adaptive approach are compared. It is argued that
the adaptive approach will outperform the more commonly used static approach based
on the wealth of research advocating the benefits of dynamic or adaptive systems.

In Chapter 7, it is hypothesized that amalgamating decisions made independently
by the agents will produce better results due to the risk mitigating factor provided
by the increased diversity of proposed solutions.

1.3 Structure of the Thesis

The thesis has been structured into four parts. Following the introduction containing
this chapter, the literature review with in-depth information about related existing
work and the system outline used throughout this work are presented. The latter in
particular gives a detailed overview of the entire final system implementation, not all
parts of which were used in each investigation however depending on its particular
aims. The third part of the thesis contains a chapter for each individual investigation
in order to delineate clearly between each research focus. These are split into an
investigation focusing on the methodologies used, an investigation into the genetic
algorithm implementation and fitness types, and investigation into static and adaptive
systems as well as an investigation into centralized decision making. The fourth part
concludes the work summarizing the individual findings from a more holistic point of
view. Each chapter is again segmented to allow more independent and more focused study of the particular topic at hand.

1.4 Definitions of Key Concepts

The following definitions are key terms which will be used for reference throughout this work, with a comprehensive glossary included for convenience in Appendix A.

**Agent (software)** An entity in a program that operates following a specific set of rules.

**Intelligent Agent** An agent which is coupled with some form of heuristic which gives it the ability to act autonomously in a directed fashion.

**Heuristics** Search methods which avoid brute force and take a satisficing approach to finding a hopefully near optimal solution in exchange for a reduced search time.

**Genetic algorithm** An evolutionary population based algorithm inspired by the Darwinian theory of natural selection and survival of the fittest.

**Financial analysis** Methods of analyzing financial data, such as technical, fundamental and behavioral analysis.

**Equity market securities** An asset entitling the owner to a share in an institution publicly listed on one or more indices allowing for trade in that market.
1.5 Summary

This investigation focuses on demonstrating the effectiveness of an agent-based approach employing a variety of genetic algorithms to a variety of methodologies for financial trading decision-making when using real-life time-series data compared to some mainstream funds and the DAX-30 index. A series of investigations will be performed that study multiple fundamental aspects to portfolio optimization problems, aiming to contribute to a better understanding of the underlying factors affecting system performance and answering a series of outstanding and unaddressed questions. The thesis is structured into separate segments, respectively focusing on one aspect at a time prior to concluding remarks drawing on findings from each study to present a summarized overview.
CHAPTER 2

Literature Review

This chapter presents an overview of existing literature in the domain of computational finance and its related fields. The chapter will first introduce some general papers on the topic prior to discussing various previously published research on agent theories, optimization problems using evolutionary systems and financial trading methodologies. Where relevant it will point to the respective investigation in Chapters 4 to 7 to establish a more specific context.

2.1 Introduction

Agent-based systems have demonstrated particular strengths in a variety of application areas, in particular in environments where more traditional approaches have shown to possess inherent shortcomings, such as remote-controlled or collaborative systems for instance [32, 90]. Their characteristics of providing high levels of autonomy, effective cooperative attributes, ability to continue operation and adapt in
response to catastrophic failures as well as prove suitable for optimization problems, have allowed continuous advances and improved operations in a multitude of areas such as behavioral studies and economic simulations among others. The study of this field has attracted significant attention, hoping to exploit its benefits further. Good introductory material can be found in the following references [145, 137, 38, 148, 76, 5], with a personal view on the history of agents nicely summarized by Wooldridge in [145].

Various investigations have explored the ability of an agent-based system to improve operational efficiency in their particular domain [81], however research into establishing more generic frameworks of agent theory are still in a relatively early stage [147, 152, 116, 143, 20, 67]. Substantial contributions in the field have been made by the likes of Jennings and Wooldridge, who have considered many different aspects of agent theory and its potential.

In the realm of computational finance in particular, agent concepts appear recurrently placed in a variety of contexts and used in conjunction with other methodologies. As the work presented here will principally focus on this, the literature review has been structured to build towards the culmination of the various fields incorporated in this work. Computational finance, in essence representing the automated application of quantitative finance principles, is a well studied field with numerous disciplines having emerged [131, 42]. Quantitative finance itself has undergone a gradually evolving process, with a good overview and historical timeline presented by
Wilmott [139]. From the early theories of random walk and related equity models by Louis Bachelier, building on the concept of Brownian motion, in 1923 Norbert Wiener accelerated the development of the necessary mathematical models on which quantitative finance would be based. This was later followed by work from Harry Markowitz in 1952, introducing what has become known as modern portfolio theory. Continuing this work, Sharpe, Lintner and Mossin presented the original capital asset pricing model in independent publications. Similar developments followed by Black, Scholes and Merton in the form of the Black-Scholes model. In the ensuing 1970s and 1980s, the models expanded to include more factors and increased in complexity. This trend has since continued unabated with a plethora of theories and models being tested on an ever increasing number of automated systems, making the field of computational finance one of immense breadth and complexity in its diversity.

This chapter presents an overview of related works from the areas of computational finance, agent-based technologies as well as evolutionary systems, first introducing some historical and background information, prior to discussing current applications and more recent research.

2.2 Agent Technology

The idea to simulate and test ideas using agent like principles exists since the conception of Artificial Intelligence in the 1940s and 50s by the likes of Alan Turing [132] and John McCarthy [80, 79], among many others. Some argue that the advent of
object-orientation has allowed an acceleration and at times increased realism through integration of these concepts [126], with a great variety of agent definitions having since surfaced. One significant area of conflict commonly arises in the generality or specificity of any definition, as more general descriptions are criticized as being too broad and all encompassing while others exclude areas that are by some considered to be typical agents and therefore in need of inclusion. An interesting discussion on this was published by Franklin and Graesser, presenting an overview of, oft contradictory, agent definitions and establishing a clear distinction between an agent and a generic computer programme, describing the source of this diversity in definitions clearly [43].

In light of this and for the purpose of this thesis, the following descriptions will be highlighted to provide an example and basic understanding of what is meant by an agent.

**Definition 1:** “An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives”, Wooldridge (2002). [145]

**Definition 2:** “An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors”, Russell and Norvig (2002). [105]

Continuing from this diversity in definitions, Jennings discussed in his work that though the above definition may be useful, certain aspects could benefit from further explanation. To quote Jennings directly, agents should be [58]:

(i) *clearly identifiable problem solving entities with well-defined boundaries and interfaces*
(ii) situated (embedded) in a particular environment and receive inputs related to
the state of their environment through sensors and then act on the environment
through effectors

(iii) designed to fulfill a specific purpose and have particular objectives (goals) to
achieve

(iv) autonomous and have control both over their internal state and over their own
behaviour

(v) capable of exhibiting flexible problem solving behaviour in pursuit of their design
objectives, and need to be both reactive (able to respond in a timely fashion to
changes that occur in their environment) and reactive (able to act in anticipation
of future goals)

In this work the entities used in experimentation could be described as exhibiting
the following characteristics:

- State. Every entity consists of its capital and portfolio, which enables or restricts
  its ability to act depending on preceding actions taken.

- Persistence. The code defining each entity is not executed on request but runs
  persistently.

- Awareness and Reactivity. The entity perceives the state and changes in the
  environment and can decided whether or not to react to these.
2. Literature Review

- Purpose. Each entity has a clearly defined aim and purpose to its existence.

Though inherently diverse, overlap naturally does occur between agent definitions. As can be seen, the characteristics of the entities employed here exhibit a partial match to all the definitions listed above. Compared to the definition by Wooldridge for instance, entities here are clearly situated in an environment and will respond to changes in the environment in order to fulfill their purpose. Looking at the definition by Russel and Norvig, entities can perceive their environment, however as discussed in Section 3.2, the actions taken by agents will have no impact on the environment. Further relating these characteristics to the discussion presented by Jenning’s, entities here have clearly defined problem solving abilities and clearly defined goals, as well as being able to respond to changes in the environment. Given this overlap in defining characteristics and the variety in formal agent definitions, the term agent will be applied to the entities presented in this work.

Agent-based applications are being adopted by industry at differing rates, with robotics in its various forms being a prime example for industrial use such as UAVs or NASA’s Robonaut for instance [83, 29, 19, 112]. In domain specific systems, where the agent does not represent the center of attention or focus of a particular study, agent technologies have become facilitators to conduct experiments and create more realistic simulations than previously possible [126]. Russell and Norvig in particular addressed the issue of the significant differences and variety in agent-definitions rather well, in succinctly stating that the “notion of an agent is meant to be a tool for analyzing
systems, not an absolute characterization that divides the world into agents and non-agents” [105]. Franklin and Graesser added that agent definitions will inherently be diverse, as the “only concepts that yield sharp edge categories are mathematical concepts, and they succeed only because they are content free. Agents ‘live’ in the real world (or some world), and real world concepts yield fuzzy categories.”

Clearly, from this discussion, it can be seen that there is no overall consensus as to the precise definition of the term “agent”. Specific to this work and as outlined in similar related works [69, 70], an agent here is therefore defined as an entity emulating a real-life trader, providing an encapsulation framework to contain both functional information with regards to its genetic make-up, as well as financial data representing an agent’s state, determining its response and interaction within its current environment. As the emphasis of the remaining document will lie in more applied agent-based studies in financial markets, the following references provide a thorough grounding in the field of agents and pointers to further reading [60, 146].

2.3 Artificial Intelligence

Artificial intelligence could be described as “all attempts to automate intelligent human processes” [62]. Artificial intelligence is said to be “concerned with getting computers to do tasks that require human intelligence” [23]. With these definitions being quite general and intentionally vague, many more specific definitions place greater emphasis on particular elements, such as aims and activities involved, or are focused
on more cognitive aspects not too dissimilar to the definitions of agents. Again, an
abundance of introductory and advanced literature exists describing all types of ar-
tificial intelligence [21, 23, 62, 105], however the focus of discussion here will lie with
the population based genetic algorithm.

2.3.1 Genetic Algorithms

Genetic algorithms are based on a Darwinian analogy to nature where the most
successful genomes in a population proliferate. The use of algorithms based on evolu-
tionary analogies was first developed in the 1960’s by researchers such as Ingo Rechen-
berg, Hans-Paul Schwefel and Lawrence Fogel among others [110, 100, 41]. Genetic
algorithms, in particular those where each genome represents a particular solution
consisting of a bit string, analogous to chromosomes, were popularised by Holland in
1975 [57]. The population there evolves over time through various selection mecha-
nisms as well as reproduction operators, such as crossover and mutation. Henceforth,
the term ‘Holland-style’ genetic algorithm is used to refer to this particular form of
algorithmic implementation. In crossover, a new genome string is constructed from
two separate genomes and where in mutation a bit from the genome is modified ran-
domly. One variation of crossover is illustrated on the left in Figure 2.1, where two
parents produced two offspring using 1-point crossover. Mutation is illustrated on the
right in Figure 2.1, where the offspring produced by each respective parent is subject
to a 10% chance of mutation for example. However, with ample room for variation
and experimentation, many new variants of genetic algorithms have been proposed and implemented in a wide range of systems, specializing to cope with particular requirements or adjusting it to more effectively address a particular problem than the traditional Holland algorithm [48]. The following text will present an overview of some of the research done on genetic algorithms and their applications.

<table>
<thead>
<tr>
<th>Crossover</th>
<th>Mutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent 1: 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>Parent 1: 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Parent 2: 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>Parent 2: 1 1 1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>Offspring 1: 1 1 1 1 1 0 0 0 0 0 0 0</td>
<td>Offspring 1: 0 0 0 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Offspring 2: 0 0 0 0 0 1 1 1 1 1 1 1</td>
<td>Offspring 2: 1 1 1 1 1 1 1 1 1 1 0 1</td>
</tr>
</tbody>
</table>

**Figure 2.1**: Examples of crossover and mutation in genetic algorithms

As an example of an alteration of the traditional genetic algorithm for application to different problems, this much referenced paper by Cobb investigated the use of mutation to adapt standard genetic algorithms to non-stationary optimisation problems [27]. In particular, she “explores the use of mutation as a control strategy for having the genetic algorithm increase or maintain the time-averaged best-of-generation performance”. The paper proposed modifying the genetic algorithm by adding a mechanism to adaptively change its level of mutation. This was based on the analogy of biological cells entering a state of increased mutation, a hypermutable state, when stressed due to environmental conditions. She argued that for station-
ary periods in the environment, mutation remains low, while during non-stationary periods mutation increased. Her results supported this hypothesis, highlighting that with an increasing rate of change in the environment, mutation levels had to increase accordingly to adapt at a suitable rate dependent on the environment’s changes. She concluded that hypermutation permits a genetic algorithm to track an optimum in a continuous state dependent non-stationary environment, allowing the genetic algorithm to conduct a more global search moving away from any current local optima, while also allowing rapid convergence towards local optima during stationary periods in the environment.

Grefenstette also addressed mutation in his paper presenting a comparative analysis of a standard Holland-style genetic algorithm coupled with triggered hypermutation to the use of random replacement in the population [49]. The paper clearly showed that standard genetic algorithms could remain oblivious to new optima emerging in the search spaces, therefore continuing to operate at sub-optimal levels. Similarly, employing triggered hypermutation would not solve the problem until a change occurred in optima close to the existing optimum, creating the trigger necessary for hypermutation to become effective. Prior to this, no change would occur. Hence Grefenstette proposed random replacement of the population at various levels (0-99%) for a more continuous and thorough exploration of the search space. The results showed that the modified genetic algorithm could track the moving optimum better, however its effectiveness required a fine balance between exploration and ex-
ploitation. In other words, a suitable rate of replacement needs to be determined for the environment, to avoid duplicating a standard genetic algorithm but also avoiding what could essentially be a random search. The impact of such design choices was similarly addressed in Chapter 6.

Mühlenbein presented and examined the application of Parallel Genetic Algorithms, defined as a parallel search with information exchange between individuals, and their advantages compared to traditional Holland style genetic algorithms [85]. In particular, he suggested that parallel genetic algorithms are based on a small number of active and intelligent individuals while genetic algorithms use a large population of passive individuals. Furthermore, he went on to describe a standard genetic algorithm as a parallel random search with centralized control, with respect to the selection and fitness determination processes. A parallel genetic algorithm on the other hand uses a distributed selection scheme where each individual chooses a mate from its own neighborhood, creating a spatial population structure. Therefore operating without any form of centralized control in a fully distributed system like standard genetic algorithms do. Other selection methods discussed are the use of subpopulations to maintain more ‘natural’ diversity, as originally discussed by Wright in a biological context [149]. He contrasts subpopulations to Fisher’s opinion, who argued that their use was not necessary due to the nature of multidimensional fitness surfaces [39]. However, Mühlenbein argued that selection of better individuals for reproduction, thereby decreasing genetic variety of the population, would increase the probability
of producing a better individual and therefore better solutions. When taken in an adaptive context, this argument contradicts most adaptive theories that argue for increased diversification to maintain responsiveness. Therefore though a parallel genetic algorithm may prove to be a successful optimisation tool, it might not necessarily be useful in non-stationary systems.

Ghosh, Tsutsui and Tanaka addressed the issue of overly rapid convergence, suggesting that quick convergence characteristic of genetic algorithms reduce the visibility of a search space and therefore will miss possible solutions [46]. They proposed that fitness is determined using two values. One introducing the concept of age in individuals as one value, also referred to as “finite lifespan” in other literature [151], and the normal measure of fitness as the other, to produce an overall “effective fitness level”. Therefore with progressing age, where with an increasing time of existence in the population the age of an agent increases accordingly, a normally optimal solution would decrease in fitness and eventually be replaced. A representation of this effect is shown in Figure 2.2, where at a constant fitness function value the combined fitness of age and fitness function continually decreases with time. This reduces the period of an optimal solution being present but does not increase diversity and just changes the solution used. Overall, their system was able to more quickly come out from a local optimum and was therefore quicker to respond to changes in the environment. As this poses a clearly desirable quality in adaptive systems, it highlights the importance in the design of fitness functions and necessitates studies in their impact on the
measured performance.

![Figure 2.2: Impact on total fitness by introducing age or finite lifespans to solutions](image)

Anunziato et al focused on developing autonomous structures able to dynamically generate optimised-control rules, as well as the acquisition of knowledge through observations and measurements rather than providing knowledge [8]. In their experiments an evolutionary control structure is employed, where continuous learning occurs through a combination of assessment of regular measurements to determine fitness, and a consequent updating of the environment for further optimisation. To do this they compared unknown situations with previous similar instances to derive a solution. However, they found it not necessary as eventually the optimal solution is derived and there are no significant improvements when a priori is used. Furthermore, they used finite life spans in agents to eliminate aging solutions. This was found to result in an erratic environment due to near optimal solutions being lost,
but also demonstrated adaptation on the other hand. In comparison to infinite life spans, performance was smoother but displayed less rapid adaptation. In summary, a direct trade off was shown between reactivity of the system and its stability.

In a more application specific paper looking at knowledge acquisition, Grefenstette, Ramsey and Schultz present a very good overview of genetic algorithms in an unpredictable environment [50]. They explore methods to automate the knowledge acquisition process to define rules to maximize the system specific payoff. To this end a virtual system is used for training that feeds back into a real-world system, with the learning component adjusting rules to improve prediction accuracy and penalizing both incorrect and overestimated predictions. The highest strength rules therefore must have a high mean value as well as low variance in its estimated payoff. Furthermore, fitness is measured by the difference between the average payoff and a baseline performance measure, eliminating agents that do not meet a base standard and thereby encouraging higher performance further. Following the principle that rules are not divisible and if both parents share one rule, all offspring will also hold that rule, causing propagation through the population of consistent high performance rules [68]. Their results however showed a disparity with previous findings by Fitzpatrick and Grefenstette, who showed that performance generally improves with an increase in population size [40]. They argued however that for larger search spaces, larger populations are necessary for sustained exploration. In line with other existing research, they also showed that a genetic algorithm employing crossover in
addition to just mutation performs better than a genetic algorithm that only uses mutation without crossover. An important point highlighted by their study is the potential problems associated with transferring simulation trained systems to real environments. In particular they showed that using a general and noisy environment for training would result in slower training, but a far more robust system that is capable of coping with such variable conditions. If for example training occurs with fixed initial conditions or no noise and is then exposed to noisy conditions, there is a sudden major drop in performance. On the other hand, systems trained under noisy conditions can easily cope with both types of environments.

Woodward addressed the issue of the definitions and differences between genetic algorithms and Genetic Programming [144]. The paper discussed the differences and similarities found between genetic algorithms and genetic programming in-depth, arguing that their actual differences are insignificant. He argued that the major differences lie in the interpretation of the representation of one-to-one or many-to-one mappings. He argued that many “empirical results found in one field will apply to the other”, such as the assertion that “maintaining high diversity in a population to improve performance” for instance.

2.4 Intelligent Agents in Financial Markets

In this section, a brief introduction to financial markets will be given, presenting some of the tools commonly used by both humans and programs in analysing financial
markets, with a particular focus on equity markets. This is then followed by a look at the increasing use of agent principles in wider research ranging from economics to artificial stock markets as well as the use of artificial intelligence in this domain. The summary will then draw together all areas and present a sample of existing research on intelligent agents in financial markets.

2.4.1 Methods of Financial Data Analysis

With a variety of approaches to analysis in existence, it is possible to classify them into “fundamental”, “technical” and more recently “behavioral” analysis. The following synopses will predominantly focus on technical analysis however. Though behavioral analysis [127] and fundamental analysis [35] may provide many interesting ideas to expand future research into, these will not be discussed here.

Starting with a paper advocating technical analysis by Brock, Lakonishok and LeBaron, they looked at whether a Moving Average and Trading Range Break are valid tools for trading, expanding the argument to technical analysis in general further into their discourse [18]. In the paper the authors cite a plethora of research providing evidence of predictability of equity returns from past returns, which contrasts the random walk hypothesis [86] for instance. This is explained through market inefficiencies or time-varying equilibrium returns. They also highlight a range of documented effects observed in data such as seasonal and weekend effects among others. The moving average and trading range break rules were evaluated based on their
ability to forecast prices. They highlighted the danger of data snooping occurring, whereby “the more scrutiny a collection of data receives the more likely interesting spurious patterns will be observed”. They suggested using non-overlapping subperiods for statistical inference as one measure of prevention. In their experiments they used 90 years of data from the Dow Jones Industrial Average (1897–1986). Several variations of each technical indicator were studied, using different short- and long-term values, as well as using each with and without bands. In their results they found returns to be predictable and showed consistency across subperiods in their dataset. This outcome is also supported by previous work on the Standard & Poor Index by French, Schwert and Stambaugh [44]. Overall, their results showed strong support for both indicators and they concluded that technical analysis helped in predicting stock price changes. However, it is important to note that no transaction costs were included in their study and that their focus was on the Dow Jones Industrial Average, not on the individual securities listed therein.

Less optimistic results were provided by Hellström and Holmström in their paper looking at trend analysis and using trends as predictor variables [56]. They pointed out that “technical prediction of stock returns is notoriously extremely difficult” [55], and therefore tried to examine the “concept of trends and how it can be utilized for predictions of stock returns”. The paper defined a trend-following trading strategy as “buying stocks which have shown a positive trend for the last days, weeks or months”. They continued to suggest selling stocks that are displaying a negative
trend. Having tested this strategy statistically, they quite notably concluded that there appears to exist no correlation between past and future price changes. They then went on to argue that a time series is sometimes predictable and that in such instances future prices can be predicted more accurately than through pure chance alone. The fact that success or fitness is usually measured based on generated profits naturally reduces the importance of a prediction’s accuracy, hence justifying the huge variety of technical indicators that exist today.

Lipton-Lifschitz discusses applied mathematical frameworks in financial markets, mainly focusing on the role of derivatives in solving financial predictability problems [75]. The paper reiterates that the prices of stocks and bonds have long ago been established as random, and that consequently price prediction would make little sense. However he still argued that derivative instruments could provide some indication of future prices despite this.

Pau described price evolution in markets using technical analysis in a graphical form [93]. He argued that it is important to note that curve interpretation in economics and finance are technically feasible in terms of theory and tools. In the experiment, he segmented curves into elements (e.g. uptrend or downtrend, peak or valley, etc.) from which trends and movements are analysed, on which trade decisions are made. This is in contrast with most other approaches where purely mathematical models are used.

Similarly, Wilson et al present a research summary into pattern recognition in
2. Literature Review

Time series, defined as repeating subsequences occurring within a time series [141]. Its novel and key contribution is the detection of unknown patterns and its neutral perspective on the data analyzed. In this work the algorithm was tested on the execution of processes within a system while it has also been applied previously in analyzing time series of oil prices with positive results [140]. Results again showed positive results with the algorithm able to identify patterns correctly, promoting it as a very robust approach to time series analysis.

As a concluding remark to the disparate views expressed in the above papers, it might be worth referring to Moody, Levin and Rehfuss, who in their 1993 paper [84] quite accurately pointed out that predictions of the economic system are difficult due to non-quantifiable factors such as mass psychology and sociology which influence economic activity, providing a possible explanation of the wide range of complex experiments and contradicting results found in this area of research.

2.4.2 Evolutionary Systems in Financial Market Applications

A literature review of 64 publications by Wong and Selvi provided an overview of a variety of neural network applications in finance from 1990 to 1996 [142]. They stated that neural network technology “was developed in an attempt to mimic the acquisition of knowledge and organization skills of the human brain”. The benefits of neural networks mentioned were of the significant support they provide in terms of organizing, classifying and summarizing data, but also discerning patterns in in-
put data, the need for few assumptions and high levels of prediction accuracy. In their survey, only a few papers developed neural networks that supported strategic planning for decision-making, as neural networks present inherent limitations in this respect. Neural networks cannot explain themselves, are inductive and require large data sets while decision making focuses on unusual and non-routine decisions. However, they do advocate use of neural networks when integrated with other technologies for decision-making applications such as expert systems for example. They conclude that a combined approach of statistical techniques and neural networks is better than either method in isolation, and foresee the future of neural networks primarily as an integrated tool.

![Artificial Neural Network](image)

**Figure 2.3**: An illustrative schematic outline of an Artificial Neural Network

Dunis and Jalilov investigated the use of neural network regression and other forecasting techniques in financial forecasting models and financial trading models [33].
Their aim was to assess whether neural network regression models add value to applications compared to simpler modeling techniques. Their trading models were assessed based on trading efficiency and not forecasting accuracy, and the data was partitioned into $\frac{2}{3}$ for training, $\frac{1}{6}$ for testing and $\frac{1}{6}$ for validation. They found that their neural network regression based model demonstrated superior performance in terms of maximizing returns and risk-adjusted profitability.

Gutjahr, Riedmiller and Klingemann provided a typical example of predicting a noisy time series [52]. Their paper studied the exchange rate between the US Dollar and the German Mark using a neural network. It was done in cooperation with an established bank aiming for a real-time end product. They introduced their material by arguing that the likely failure of past prediction models is due to their linear nature, but the non-linear model approach of neural networks appears adequately complex. Preliminary findings were that a standard neural network did not produce good results; hence they proceeded to concentrate on individual issues in economic time series. This was done by selecting patterns for the neural network to learn, as this would help the model understand the market better. Three criteria for the following pattern definitions were used:

- Soft-trend. Time horizon of 20-80 days.
- Trend. Time horizon of two days to one week.
- Jump. Where change is above a limit, suppressing fluctuations and allowing
analysis of only substantial/significant trends/movements.

With this they found that “the selection of training patterns is a sensible way to improve the performance of a neural network”. They also concluded that more data did not lead to increased performance, an issue also addressed in Chapter 6. Using daily buying and selling it performed “significantly better than random walk”, an important finding as it indicates a good forecasting system suggesting that the application of strategies outperforms a simple buy and hold approach.

The following paper by Kohara et al takes the unusual approach of attempting to integrate both quantitative as well as qualitative data into their experiment looking at price prediction of securities [66]. In their price prediction they included non-numerical factors such as political and international events, extracting qualitative information from newspaper headlines, referred to as event-knowledge. This was then used only to show tendencies in price movements. They found that approximately 60% of the time this corresponded to the actual next-days movement. Trade signals are generated based on the predicted up or down movement of the main Tokyo stock exchange when this value was found to be higher or lower than a pre-set benchmark. No transactions costs were included in the simulation. The results showed that there were fewer errors than using a multiple regression analysis with 5% significance.

Pi and Rögnvaldsson presented a futures trading system that employed a non-stationary neural network prediction model, looking at short and long-term trends [96]. Their trading strategy took the approach of constructing a model that forecasted
price movements and subsequently generated trading signals based thereon. In this instance, they employed a non-linear prediction model for finding market trends and a filter system for selecting the most suitable indicators to generate trading signals. Quoting Weigend and Greshenfeld in that “feedforward networks have been applied to predicting actual prices or price changes with little or no success” [136], they choose to not forecast prices but rather look for trends and changes therein. An interesting feature of their system was its adaptive nature, where for every tick the model is retrained on the same old data plus the additional data from the last tick, updating the model for the current market environment. They argued that it seemed like non-linear prediction models and filtering strategy could pick up price movements and consequently generate positive excess returns. They also pointed out that their ‘retraining’ strategy after every tick helped overcome the nonstationary problem in financial data series as well as make more efficient use of the available data. This benefit of adaptive systems in particular is studied more extensively in Chapter 6.

Toulson and Toulson present a forecasting and trading methodology for financial markets again using neural networks [129]. They employed Markowitz Analysis to manage risk and maximise portfolio returns, which requires accurate estimations of expected returns that were obtained using weighted averages of historical returns. They proposed the use of neural networks to estimate the future returns and risk, arranging these into committees of neural nets that then managed the portfolios. Prior to running their system, they manipulated the data to reject outliers using a
seven standard deviation threshold applied to equity returns, replacing them with simple interpolation. This process smoothed the time series and thereby eliminated some fundamental characteristics of the series, in turn possibly affecting the final outcome. However, they argued that the input reduction technique achieved a performance increase for accuracy. For prediction, they employed multi-layer perceptrons (returns prediction and volatility), k-nearest neighbor (returns prediction) and stochastic volatility (volatility) models. These were linked into a committee structure of networks, predicting seven days in advance. The performance of committee networks proved to be slightly better than that of single multi-layer perceptrons and much better than the k-nearest neighbor model. For volatility prediction, little difference was found between the committee network and stochastic volatility model with a slightly reduced prediction error in the network.

Contrasting their approach of data mining algorithms for a portfolio trading system to neural networks, Lemke and Müller modelled self-organization and discussed the use of the Group Method for Data Handling [72]. In contrast to neural networks, the Group Method of Data Handling works on the principle of induction. They suggested that a self-organizing fuzzy-model using this method might be useful for badly defined financial applications such as, for instance, generating fuzzy signals for a portfolio of data. Interestingly, the paper also discussed the disadvantages of neural networks in financial markets. For example, they argued that knowledge is inherently hidden in neural networks and thus cannot be used for data analysis, interpretation or
validation without using extra tools. Furthermore, they argue that prior experience of the developer is needed in setting up the structure of the network, which the internal representation of knowledge does not support. Without validation a neural net could easily be affected by over-fitting when using noisy data samples, which might lead to a sub-optimal model. To generate trading signals, they used a modified Moving Average Convergence Divergence indicator. They argued that trading signals would by definition lag behind the optimal trading point, representing lost potential profit, a well acknowledged fact in technical trading. Consequently, they suggested minimizing this effect by considering both historical and predictive information. For example, a 20 day moving average convergence divergence indicator would span 14 historical days and 7 days based on predicted prices. This would then allow for improved profit maximization. However, the validity of the prediction mechanism would introduce an additional element of uncertainty, thereby arguably reducing the usefulness of the observed signals. This would therefore introduce another trade-off, whereby reduced lag is exchanged for heightened uncertainty of the indicators output.

Doeksen et al focused on the prediction of price movement in individual stocks using a variety of artificial intelligence techniques [31]. A combination of neural networks, genetic algorithm and fuzzy inference systems were used to investigate the profitability of directional price movements. They used a data sample split into training and testing data, with data normalized for better compatibility with their system. Results showed that their system was able to learn the historical patterns well, as
other research suggested previously using similar methods. Overall the authors suggest their preliminary work can be built upon to develop predictive systems with a larger degree of reliability. Given the existing drawbacks of their current strategy, however, such as continuous quoting resulting in high costs due to fees, makes their current approach suitable for a simulated environment only.

Presenting a good introduction to automated trading in general, Dempster and Leemans studied an adaptive reinforcement learning algorithm in the foreign exchange market under considerations of risk as well as comparing their adaptive system design versus a static simulation run [30]. In their work, an additional optimization overlay to fine tune parameters used within the evolutionary system itself was superimposed in reaction to changes in the underlying market. Their findings showed that the element of risk in particular would affect performance significantly, as with increasing aversion performance would drop while the profit per trade would increase noticeably. With respect to their adaptive system design, they compared it to the performance resulting from optimizing system parameters over both training and testing data, producing optimized settings with the best fit for that data set. Despite this being an arguably difficult benchmark to attain, their adaptive approach proved superior. This further supplemented the argument that despite static systems being able to evolve robust solutions that will perform adequately over time, the adaptive approach can maximise returns and adjust to ongoing variations in the environment. The work presented in Chapter 6 addresses the same question, as well as providing an in-depth
2. Literature Review

study of the outstanding issue of preceding system optimization, by contrasting two purpose built adaptive and static evolutionary systems, based on otherwise constant parameters for fairer comparison.

Ghanda et al investigated the evolution of an adaptive stock selection system using fuzzy logic based trading rules [45]. In their evolutionary system, both mutation and crossover were employed, while also introducing a repair operator. This operator essentially monitored new solutions produced for their degree of similarity to the currently best solution, eliminating solutions that diverged significantly. Following the often advocated benefits of maintaining population diversity in a solution space, this operator effectively guarantees conformity to one particular solution template prior to even evaluating the fitness of new solutions. They also briefly discussed three methods of evolving the trading rule base, representing a static and two slightly different variations of an adaptive approach. The difference in the approaches rested in one being retrained over all historical data available while the other merely used a much smaller sample of 120 of the most recent days. However, they only actually appeared to use the adaptive approach, based on a smaller set of recent historical data, without ever testing the other alternatives presented on their own rule base. Furthermore, they earlier suggested that the adaptive approach simply appeared more promising than the static approach. Though a very interesting paper, it is surprising that no comparison appears to have been done on the differences in static versus adaptive systems, further highlighting the need for clearly documenting the benefits
and drawbacks of either approach in a comprehensive examination, as presented in Chapter 6. Their findings showed that the fuzzy rule base was highly successful, significantly outperforming the other methods they implemented for comparison. It needs to be highlighted that some of these other methods used were partially based on the static approach. In their concluding remarks, they attributed this noticeably superior performance of their own rule base to their use of the adaptive approach. However, they did not perform a direct comparison of static and adaptive approaches.

Earlier, Yan and Clark had presented results supporting the importance of maintaining diversity and its positive effect in achieving higher fitness and adaptivity [150]. They presented a new genetic programming algorithm, in which population diversity is enforced by removing all solutions, bar one, that have a very high correlation to one another. In this way, they attempted to address traditional limitations highlighted by the work presented in Chapter 6.3 (published in part as [106]), where population diversity is maintained using restart methods or immigration of new solutions. Tested on 41 months of data consisting of end-of-month information on each stock, they found that not only was population diversity maintained, but the best solutions created by their genetic programming consistently outperformed that of solutions evolved by a standard genetic programming implementation. However, it should be pointed out that the difference in mean fitness of solutions between both approaches remained insignificant. This contrasts, noticeably, with the repair operator just mentioned above as part of Ghanda et al’s work [45].
Grosan et al on the other hand introduced a novel genetic programming approach, referred to as multi-expression programming, in order to predict the movement of two indices [51]. To establish a relative comparison, results of their error minimization were compared to a number of alternative intelligent approaches such as a neural networks and fuzzy models for example. Results obtained suggested that their novel technique offered a good candidate for future research and minimized their fitness function successfully. They further argued that based on their work, it appears that one paradigm alone would struggle to perform well for multiple indices, perhaps suggesting a need for the wide diversity of algorithms, systems and paradigms used in financial market applications.

The work of Feng et al looked at a market making strategy that focused on exploiting market volatility, compared to a technical analysis based strategy focused on exploiting directional movement of prices [37]. Both strategies were tested on the Penn Lehman Automated Trading simulator [61]. Results were mixed and varied largely depending on the data presented, suggesting a lack of adaptability, as well as a tendency to neglect risk and assume large positions. Overall the work performed well in the live competition however and allowed for later further work by Sherstov and Stone, where the performance of 3 approaches to autonomous stock trading agents were again tested in the Penn Exchange Simulator [117]. They used a reinforcement learning based agent employing machine learning methodology focused on long term performance, a trend following based agent based on price prediction using linear
regression, as well as a market making based agent based on capitalizing on small fluctuations in prices producing a more consistent return in performance. Overall they found that only their market making based strategy managed to return positive results as measured using the sharpe ratio in this run, while showing strong potential following further refinement for all strategies presented.

In the work of Tanaka-Yamawaki and Motoyama the focus lay on the prediction of very short-term movements in a time series, referred to as ticks, which can be divided into up, down or unchanged [124]. The authors studied a time series of the US Dollar to Japanese Yen and using an evolutionary algorithm measured the frequency by which a solution was able to correctly predict the next tick movement. Noticeably, predictive accuracy increased when they increased historical tick data from 1 to 2 and 2 to 3 past ticks, however this effect plateaued quickly. Further, the information deduced from the number of historical ticks used was expanded from simply taking the average of previous ticks to determine future direction to calculating conditional probabilities using historical tick movements. Their best results returned an accuracy of 70%, where the authors suggested that the inherent random nature of the time series might represent a ceiling level leaving no further room for improvement possible on this result. The implication of any accuracy above 50% naturally suggests a significant profit margin on any trading approach utilizing this strategy, likely to be replicated quickly within the market place, diminishing its effectiveness. Though whether a ceiling in terms of accuracy of predictions exists, and at what level it may
2. Literature Review

be, is not substantiated conclusively and remains an open question.

2.4.3 Genetic Algorithms in Financial Market Applications

A key paper on using genetic algorithms and technical trading rules was first published by Allen and Karjalainen [4]. The authors used a genetic algorithm to evolve technical trading rules using daily closing prices at various transaction cost levels, and found that these rules did not earn consistent excess returns over a simple buy-and-hold strategy. They stated that there had been little formal analysis in financial economics of genetic algorithms and that the purpose of their paper was to demonstrate how genetic algorithms could be used to find technical trading rules. A significant amount of literature exists on the effectiveness of technical trading rules, which found that these did not make money, especially when including transaction costs. Fama even went so far as to dismiss technical analysis as a futile undertaking [36]. These findings were supported in their research, where they said “it is not possible to make money after transaction costs using technical trading rules”. In their system, they used a genetic algorithm to explore both the structure and parameters of the rules at the same time. It functioned by taking individual trading rules and recombining them to more complicated ones, representing crossover. Fitness is measured on excess returns over the buy-and-hold strategy. They conducted the experiment using three levels of transaction costs, 0.25%, 0.5% and 0.1%. They observed that by creating a portfolio of trading rules they allowed for more diversified risk and reduced volatility of 8.7%
2. LITERATURE REVIEW

in this case. Obviously, they also found that high transaction costs led to reduced trading frequency and vice versa. They found that low transaction costs led to higher returns and better forecasting ability, while high transaction costs led to reduced forecasting abilities. In other words, returns are sensitive to transaction costs. The paper also discussed the growth of trading rules used, and how they had “a lot of redundant material”. Overall they found that “trading rules do not earn consistent excess returns after transaction costs. Nevertheless, they appear to have some ability to forecast daily returns”. However they also pointed out the possibly distorting effects as a result of the lack of dividends in their simulation, which in actual markets have a well documented impact.

Following on from this work, Neely published a paper discussing the risk perspective of Allen and Karjalainen’s system [88]. Essentially, they found that there was indeed no evidence that their evolved rules outperformed buy-and-hold strategies on a risk-adjusted basis and were therefore consistent with market efficiency. They argued that risk adjustment is “absolutely essential” for evaluating usefulness of trading rules and measuring consistency of results with market efficiency. In their simulation, interest was earned in terms of calendar days and not trading days, as this corrected inaccuracies in Allen and Karjalainen’s work. They pointed out that “to determine whether technical trading rules can produce better risk-adjusted returns than the buy-and-hold strategy, ideally we must train a set of rules using the Sharpe ratio as the fitness criterion.” They also used the X* measure as proposed by
Sweeney and Lee [123], suggested to be more appropriate for equity markets. Here, positive $X^*$ statistics are interpreted as evidence of superior risk-adjusted returns. Out of all three fitness criteria (Sharpe, $X^*$ and Jensen’s $\alpha$ [59]), $X^*$ produced the most predictive content. Overall they found that though risk adjustment improves the relative attractiveness of the rules, none might have been useful to risk-averse speculators. Furthermore, the inclusion of dividends, removal of spurious autocorrelation or accounting for price slippage would strengthen the negative results of this exercise.

Becker and Seshardi described a method, also based on Allen and Karjalainen’s work, to evolve a set of technical trading rules in form of trees using genetic programming, when dividends are excluded from stock returns [13]. An example of a tree structure used in genetic programming illustrated using a simple mathematical function, the expression $7 \times (2+1) + (8 - 1)$, is shown in Figure 2.4. The major changes made were the use of monthly rather than daily data, reducing the operator set and increasing the number of derived technical indicators, using a complexity penalizing factor in the fitness function, utilizing a fitness function that considered the number of periods of good performance and finally, using co-evolution of specialized buy and sell rules. In their experiment they used a population of 500 over 100 generations, with a transaction cost of 0.5%. Three experiments were run, with the first using a fitness function including a cost-penalizing factor, the second where fitness was the number of well performing periods while the third used co-evolved specialized trading
strategies where buy strategies mated only with buy strategies and sell strategies only with other sell strategies. In their first experiment, they noticed overfit occurring for rules without the cost-penalizing factor and that rules with the cost-penalizing factor outperformed both in training and testing. However, what needs pointing out is that a substantially lower number of trades per period resulted in a near 'buy-and-hold' strategy compared to Allen and Karjalainen’s work, possibly suggesting that improvement in results is due to longer holding periods rather than any other changes. With a clear superiority of the buy-and-hold strategy persisting in this system, a fall in the number of trades due to longer holding periods would therefore obviously lead to an increase in returns, moving ever closer to a buy-and-hold pattern.

Potvin, Soriano and Valée used genetic programming to evolve trees of technical trading rules [98]. They used two types of trees that ‘grew’ or were of a ‘fixed’ size, while reproduction methods included duplicating existing trees, crossover replacing a branch of a tree and mutation, replacing a branch with a newly randomly generated branch. Selection occurred through ranking, where their raw fitness values were translated into a standard ranking for fitness-proportionate selection. Fitness was defined as the excess returns over a buy-and-hold strategy. Data used was 14 securities from the Toronto 300 stock exchange over 2003 days, with the system starting at day 250, as some rules used data from 250 days back. They also used two datasets, one with a short training period and one with a long one. The best parameters observed were selected manually, it appears, for use in the experiments. In the analysis of
their results, they argued that over-fit might have occurred but reducing generation
numbers did not seem to affect this. Overall, slightly negative returns were observed
though the majority of securities showed a positive return compared to buy-and-hold.
They concluded that their trading rules are good during stable or falling markets, but
in appreciating markets the buy-and-hold approach fares better. This representing a
positive beta factor (positive correlation value) of $0 < x < 1$, where the performance

$$7 \times (2 + 1) + (8 - 1)$$
of the trading rules follows that of the securities price, at a reduced magnitude.

In a study by Setzkorn et al, they used genetic programming to optimize a trading rulebase. The rulebase representing technical indicators, in this case only including the Moving Average, which can produce buy/sell/inactive signals. The rulebase was then applied to the S&P composite index (2.1.90–18.10.01) [111]. Two systems were developed, one using few rules and nodes and the other with higher complexity. They were based on a static genetic programming rule system, with selection only being determined by the trading system with a fixed duration investment. Tournament selection and 1-point crossover were employed by the genetic programming. In comparison to the buy-and-hold strategy both failed to outperform it. In their conclusion they found that the complex rulebase suffered from overfit, while the simpler one, though more robust, failed to capitalize on market trends.

In the work performed by Streichert et al, the impact of three different crossover operators using a multi objective evolutionary algorithm on the portfolio selection problem was examined [122]. The results showed few differences between the different crossover operators, which the authors argued must have been leveled out due to reduced search spaces or due to the speed of convergence in the system. Primarily, their work explored the use of multi objective evolutionary algorithms as an initial excursion into its application and as preliminary work to encompass further real world constraints that may affect portfolio selection.

Korczak and Lipinski studied the use of a genetic algorithm inspired evolutionary
algorithm to evolve traders on historical stock data, contrasting the use of a large set of 350 trading rules to a smaller set consisting of 150 of their linear combinations [74]. One of the leading arguments presented was the benefits achieved through the significantly reduced computation time when using the trading rules’ linear combinations, enabling their application for real-time systems. In their experiments, following training, the traders were exposed to testing data and results showed that a significant reduction in system time for the linear combination approach could be achieved while demonstrating a negligible loss in efficiency compared to the larger set of trading rules. In conclusion, this paper argues that a reduced set of rules can provide equally positive results while reducing the system’s resource consumption, thereby allowing its use in adaptive systems to take advantage of the benefits offered by these in turn.

2.4.4 Agents in Financial Markets

A very good overview of agents in financial markets was written by LeBaron, who presents an overview of research done in artificial markets using agent-based approaches from its first appearance up to 1998 [69]. He argued that financial markets are an important application for agent based modeling styles and offer many appealing characteristics, a notion that is obviously shared by many based on its wide usage. For one, he argues that data is easily available in many different formats, from minute by minute to monthly, and information is often ‘sharper’ with agent objectives being clear and relatively straightforward. First he presents a paper by Lettau, who imple-
mented evolution and learning in a trader population and created a basic benchmark for comparison. In particular, it raised the basic yet remaining crucial question of how to determine fitness values. This in particular will later be addressed in this work in Chapter 5. The next paper by Gode and Sunder investigated the level of intelligence required by traders to generate positive results [47]. In their results they showed that random traders are generally inefficient compared to their other trader sample, which with reduced randomness performs at levels comparable to humans. Among the other papers discussed, Arifovic’s differed through the use of an ‘election operator’, where offspring in the genetic algorithm only replace their parent if their fitness was at least equal to that of the parent on the last test run. With a focus on uncertainty and information in markets, another paper by Routledge concluded that with high noise levels, forecast parameter precision is less crucial compared to situation with low noise levels, hence affecting their learning [104]. LeBaron also discussed the Santa Fe artificial stock market, where he defined it as a model that tries to combine both a well defined economic structure in the market trading mechanism, along with inductive learning using a classifier based system. In this system an agents individual assessment of the market using the classifier system determines the state of the economy and forms a price/dividend forecast which goes into the demand function of the system. This should therefore provide a detailed understanding of that market’s dynamics. In other studies Youssefmir and Huberman found that resource allocation and prediction of other agent’s actions substantially affected decision-making [151].
They found that the choice of learning method often heavily influenced the results obtained and the artificial stock market model created. The paper also mentioned the difficulty in formulating a fitness function that emulates the human goal of long run wealth maximization accurately, rather than relatively simple utility maximization. They also argued that there still remained a lack of clarity whether agents with finite lifespans specific to an environment or more robust agents with infinite lifespans are better suited for representing financial markets.

Following on from the notion of artificial stock markets, Ankenbrand and Tomassini attempt to realistically simulate multiple interacting financial markets using populations of artificial traders and investors, capable of trading in all markets simultaneously, hoping to increase understanding of the behavior or financial markets [7]. The originality of their work lies in its multi-market nature, as they argue that “real markets are strongly correlated”. Inter-market effects were therefore included in their simulation. They mention that a major factor to be taken into account in realistically simulating markets “issues of beliefs and prediction about the future become important and in the individual beliefs and choices when aggregated shape the economic indicators, the market prices and ultimately the world that the agents must deal with. Moreover, this world is dynamical, as a result of the constantly changing collection of beliefs and strategies of the agents striving to adapt to the system evolution”. While discussing technical traders, they argue that technical analysts “believe in a trend behavior of the markets and try to use moving averages or momentum analysis for
2. Literature Review

detecting trends”, where they then go on to state that the disadvantage of “trend theory is that traders buy assets when they are high or expensive and sell when they are low and cheap”. On the basis of the latter statement, it is an obvious conclusion that this implies a fundamental loss making strategy, which is not the case as has been shown in the past, as technical analysis can be effective, both for real and simulated traders. They also pointed out the need for agents to be adaptive, and suggested the use of neural networks for future work.

A more general paper on artificial stock markets, Boer, de Bruin and Kaymak introduce a framework for experimentation and the use of intelligent agents for modeling asynchronous behavior and continuous trading [15]. It highlights the need for artificial stock markets to realistically represent real markets, as both structure and environment are the determining factors in market dynamics. It reasons that artificial stock markets are created to improve understanding of market dynamics and most provide a useful tool for testing hypotheses. Understanding is improved by attempting to reproduce market movements or looking for successful strategies, comparing fundamental versus technical approaches. They also point out that superiority of technical analysis over fundamental analysis seems to indicate basic market inefficiencies. Furthermore, due to constantly changing market conditions agents need to adapt through learning by evolving new strategies. The paper also touches on the issue of most artificial simulated markets creating only investors, and none of the peripheral entities that regulate or also interact with the market such as market makers
or brokers, who act not on their own interests but on behalf of others, leaving it an incomplete environment.

Though far more papers exist on this topic [73, 91, 119, 126] among many others, the above are rather good introductory texts.

2.4.5 Summary of Intelligent Agents in Financial Markets

Van den Bergh et al presented an overview of the use of agents for studying market environments [134]. They argued that markets could be considered multiple auction environments to determine prices for the assets traded in the market, which can therefore be equated to distributed price determination. They point out that in order to study complicated forms of interaction, agents themselves must possess a rich cognitive structure. In their model they used an agent that could affect the environment to a small degree, which they argued parallels the definition of a decision maker in a decision environment making this a decision-theoretic approach to agents. They describe four levels of adaptation in an agent:

- Weak adaptation - where an agent can modify its environment but not itself
- Semi-weak
- Semi-strong
- Strong adaptation - where an agent can change its intentions (functional adaptation) as well as environment
They concluded that their model would assist in analysing financial markets for behavioral characteristics and result in increased understanding of market dynamics, possibly leading to policy being able to regulate markets for the maximisation of global wealth.

In 2003, Kearns and Ortiz introduced the Penn-Lehman Automated Trading Project, with one of its primary motivations to contribute to developments in automated market system and competitions [61]. It distinguishes itself by focusing on automated markets and strategies. They are particularly interested in “developing clients that make predictive use of limit order book data, including those using statistical modeling and machine learning”. The centerpiece of their project is the Penn Exchange Simulator, a “software simulator for automated stock trading that merges automated client orders for shares with real-world real-time order data”. It is an Electronic Crossing/Communication Network, whose fundamental role is the “computerized maintenance of buy and sell order books in offered stocks, automated order execution and various other related functionalities”. In the project, clients can sell short, in other words sell more shares than they hold, or buy shares without cash, with their valuation being the sum of their capital holdings and share holdings at the current price. In other words, it assumes infinite liquidity. In their competitions they have thus far found a variety of strategies, ranging from market-making and technical trading approaches to “new methods relying heavily on order book data”, all of which were tested on one stock, namely Microsoft. The results showed that 11 of 14 clients
ended with overall positive earnings for a 10-day competition. They also found that their internal Penn Exchange Simulator market price correlated fairly closely to that of “Island”, their reference electronic crossing/communication network.

Earlier work in this area of looking at high-frequency order book based trading was performed by Cliff, who studied an extension to the Zero-Intelligence traders introduced by Gode and Sunder in [47], referred to as Zero-Intelligence Plus traders [26]. In his series of investigations, agents were placed in a typical financial market environment with order books resulting in a bid-ask spread, also known as a continuous double auction environment. These Zero-Intelligence Plus agents place orders into the market at a certain limit, with a number of parameters determining this limit. The optimization of these parameters is performed by a genetic algorithm. Though initial results showed the agents outperforming their human counterparts, based on a configuration of 8 different parameters, later work extending these to 60 parameters further improved results significantly [25]. Though these studies were applied to order book based trading, a comparatively short-term perspective on financial markets compared to the investment strategies investigated in this work that are not influenced by bid-ask spreads, the successful optimization of trading agents using a genetic algorithm encouraged the decision to apply these evolutionary principles in the work presented here.

Pictet et al discussed the application of genetic algorithms to deal with the sharp peaks of fitness that are usually not representative of a general solution in financial
problems, but rather indicative of some accidental fluctuations [97]. In particular, their argument aimed at avoiding overfitting, defined as building indicators to fit a period of past data so well that they are no longer of general value. To minimize this, they argued the following elements would have to be present:

- A good measure of trading model performance
- Indicator evaluation for different time series
- Large data samples
- Robust optimisation techniques
- Strict testing procedure

In other words, they argued that if the above criterion was maintained, it should result in a more general and robust system that does not converge to one solution to such a degree that it makes it inflexible and overly specific. Interestingly, their trading system employed an approach whereby antisymmetric indicators, momentum of the price logarithm for example, are used to provide the dealing signal, while symmetric indicators, such as momentum of the absolute price change, modulated it. For instance, a symmetric indicator could alter threshold values and prevent a trade or vice versa. Another argument put forward was the difficulty of optimising trading models due to the noise present in the data as well as the risk of overfitting, whereby the challenge is to find indicators that are robust in the sense of being smooth and
giving consistent results out of sample. Agents are trained in three phases on three separate sets of data, with the first being a build-up of the indicators, the second for optimization of model parameters and the third for selecting the best trading models. They presented an in-depth study of genetic algorithms applied to noisy, discontinuous and complex search landscapes, presenting a thorough argument for their use.

Schulenburg and Ross explored the degree of reliability that intelligent agents can have when applied to real life economic problems using a learning classifier system [109]. Since understanding of precisely what it is that generates financial time series is still limited, they suggested an alternative approach and sought to model agent behavior rather than time series, as they argued it to be one of the major factors influencing market behavior. Their experiment was based on making different information available to three separate types of agents to make their decisions on, hoping to more realistically reflect diversity in market participants. The learning stage of their experiment was split into two phases, with the first determining which strategies were useful, while the second was modified using a genetic algorithm replacing the worst rules and creating new ones based on existing successful versions. Overall they showed that all traders were able to outperform the buy-and-hold strategy as well as the bank investment benchmark, and argued that technical trading is a valid outcome in this market, with informed agents performing better than buy and hold agents for instance. They also raised the question of whether a system of
agents could emulate real financial data using evolutionary methods to configure the system’s parameters. In particular, in order to successfully and realistically emulate real market data this would imply emulating all the various factors affecting market development such as expectations of market participants, which in turn are determined by mass sociological and even individual factors. This could be viewed as a rather ambitious goal. Regardless, the investigation was a successful demonstration of an intelligent trading system and presents good results and further supplements the case for increased use of technical analysis based systems to operate in markets.

Looking at generating trading profits from a more global approach, Haefke and Helmenstein proposed a trading strategy that exploited informational differences that follow from different construction principles used in stock markets indexes [53], a piece of work distantly related to [7]. In this case, trading signals can be generated using two indexes offering distinctly different information to investors. Essentially this approach could be argued to take advantage of information representation discrepancies. Their paper showed that using a “neural network forecast of the arithmetic and geometric average” could generate better returns than a buy-and-hold strategy. They proposed a conservative rule using arithmetic and geometric means, where when the “slope of the geometric average intersects the slope of the arithmetic average from below, we know that the price of the low-price stocks is growing faster than the price of the high-priced stocks, and hence should buy low-priced stocks. If the slope of the arithmetic mean intersects the slope of the geometric mean from below, buy the high priced
stocks and sell the low-priced ones”. They define low-priced stocks as “stocks with a price lying below the geometric mean” while they define high-priced stocks as “stocks whose price is greater than the arithmetic average”. They found that exploiting information from comparing Indexes did indeed allow them to outperform the buy-and-hold strategy, and that through forecasting they managed to reduce the number of trades and their associated costs, thereby increasing their returns significantly.

Neely, Weller and Dittmar used genetic programming to identify optimal trading rules using two data sets for training and out-of-sample testing [87]. They mistakenly referred to Allen and Karjalainen’s paper as having demonstrated excess returns over a buy-and-hold strategy including transaction costs, while Allen and Karjalainen showed the opposite [4]. They also acknowledged that both linear and non-linear forecasting models perform rather poorly out of sample. In their experiment, they used a population of 500 agents to reproduce creating offspring with increased probability depending on their fitness in the population using crossover. They also argued that to combat over fitting of the rules as indicated by high trading frequencies, it is advisable to implement higher transaction costs to prevent this. They then demonstrated that this added ‘friction’ did contribute to a reduction in the number of trades and therefore protected against over fitting. As with most genetic programming approaches, they found their rules to become extremely complicated and highly redundant.

Kendall and Su presented a study on the use of social learning to evolve a set of trading strategies for use by heterogeneous adaptive agents to successfully trade on
one share [63]. They produced a system whereby a central pool of technical indicators is drawn upon or added to by agents to formulate their personal trading strategy. The learning approaches used for this were a neural network and a genetic algorithm, though their actual implementation remained moderately unclear, seemingly based on a hybrid approach of the two. Fitness for particular strategies was measured by taking the rate of profit, which is based on comparing current assets to that a week earlier and placing them in relation. Results comparison was done by comparing the set of agents to the price development itself and a fixed investment based on their starting capital at a fixed interest rate of 5%. In their results, most agents followed a distribution close to the share price and displayed a high correlation factor. Overall, the study indicated great potential in the use of agents in a financial context, in particular for the sharing of information within a system.

Following from the above, Kendall and Su presented a more in-depth view of the same problem, introducing a more realistic element of having a choice between five shares as well as providing more information through their social learning system [64]. It defined its aim as “essentially the process of artificial agents searching for the optimal trading strategies under certain market scenarios”. A core element to their research being the social learning, they argued that each particular type of trader will be constrained to only a certain set of information from the market, and that social learning facilitates escape from any local optima created. An interesting side-point raised was that “it will not be sensible to use a crossover operator in the genetic
algorithm”, based on their implementation thereof. However, other literature argued that crossover, rather than mutation, is a cleaner and more effective implementation of a genetic algorithm [50]. Lastly, one further point in their model is the assumption that any buy or sell action is absolute, meaning that when purchasing a share, all funds are invested in it, while for a sale, all holdings are relinquished.

2.5 Decision Making Theorems

A multitude of research exists studying and analysing the behavior and relationships of humans in group contexts, such as [34, 82, 128, 153] who looked at decision support systems and [14, 89, 92] who studied other aspects such as decisional guidance or the simulation of agents in decision making. The following are of particular interest for this review.

Stirling, Goodrich and Packard stated that excessive computational difficulty, modeling complexity and uncertainty make a complete search of possible actions and their associated consequences impractical, in particular in multi-agent contexts [121]. They stated that the main reason for choosing an action and taking a bounded rational decision is because it represents an acceptable compromise between desired performance and cost, which by definition is “good enough”. The difficulty with this is that if desired levels are set too low, performance may unnecessarily be lost, while if set too high, finding a solution may become impossible. In their paper they proposed “a new definition for satisficing, or being good enough, that applies to both
single- and multi-agent situations”. They proposed to rely on rationality that does not depend on optimisation, however at the same time taking care not to degenerate into making decisions that are not measured in terms of quality but in terms of convenience. They then argued that the comparative paradigm was underrepresented in literature as a basis for viable decision making, compared to the positive paradigm, which manifested itself in the form of heuristics, rule based decision systems etc. Hence they propose an internal comparison, where if a decision’s gains equal or exceed the losses incurred, it is defined as intrinsically rational, making it a local rather than global form of comparison. They concluded that “in complex and uncertain environments, it is essential that decision-making procedures be parsimonious and robust. The comparative paradigm accommodates both of these desiderata in a natural way. By softening the demands of strict optimality, the door is opened for a flexible and economical way of characterizing multi-agent behavior while not abandoning demands for acceptable performance”.

Collings et al focuses on automated contracting between suppliers and manufacturers using multiple criteria in a multi-agent infrastructure [28]. The main component in their experiment was their use of the Expected Utility Theory for agents to assist or guide human decision making, as full automation was unlikely due to the difficulty in modeling human notions of utility and risk. However, their argument was based on the assumption that human perceptions are the key determinants and best solution, rather than considering the possibility of a fully automated system outperforming
It also explained the expected utility theory as being a construct employing probabilities to create a utility curve, where trade-offs between wealth and utility are established. It also argues that an agent will seize an opportunity if its expected utility is higher than the utility it receives by not taking it, arguing that this is representative of risk averseness, though it appears little thought was put into elaborate existing research done in risk related analysis where a simple utility curve cannot properly account for potential benefits and drawbacks. For instance, potential profit is world domination while the drawback is forced legal closure. In expected utility, this might rank very well, while the risk is unacceptable and probability might not account sufficiently for it. In terms of communication, they operated on time-slots of advertising and bidding, but unlike other studies they suggested the use of explicit infrastructure and facilities to manage the negotiation protocols. Overall the paper provided some insight into automated contracting, but did not fully explore some of the aspects addressed within.

Vahidov and Fazlollahi proposed a framework for a pluralistic multi-agent decision support system, introducing and investigating the effectiveness of the framework for pluralistic multi-agent decision support systems [133]. Their aim was to lay the foundation for the development of an agent-based decision support system that paralleled the human problem solving process. They justified the need for decision support through an increased employee empowerment, heightened requirements for speed and quality in managerial decisions and increased accessibility to a vast amount of infor-
2. LITERATURE REVIEW

Information through electronic networks. They argued that autonomy, reactivity, social ability and proactiveness of agents could facilitate active decision support. In their pyramid model, they proposed three groups of agents that essentially formed layers to the top of the pyramid, which represented the decision maker. These layers were split in ascending order into the intelligence group, the design group and the choice group. This split allowed a gradual distillation of information to be presented to the decision maker to support their efforts. They also stated that to cover basic values, all basic standard sets of objectives should be pursued in a multi-agent decision support system, which in this case were a risk-loving agent, a risk-averse agent and possibly a risk-neutral agent. The different teams had to incorporate a range of aspects and views, as for instance two schools of thought, such as fundamental analysis versus technical analysis. They found that using such a method it outperformed traditional decision support system.

Rehfuss, Wu and Moody compared four alternative approaches to combining forecasts into one overall prediction or decision based on a sample trading system [101]. To increase robustness and adaptivity of their system they introduced retraining of their models at each step as well as a form of exponential decay to include a time dependent factor. The first system was based on a combination of all individual predictors, where if the forecasted number deviated from the mean of its standard deviation, it would generate a trading signal of the corresponding sign. The quantized predictor followed the same methodology as the combined predictor above, except with individ-
ual trading signals added. If the sum exceeded a threshold, for example two existed for neutral, short or long signals, this signal was generated. Also, the bagging predictor was based on a simple voting system whereby the signal that was generated the most was chosen. Lastly, a mixture predictor was used, where the combined forecast was added to the quantified predictor, with two votes rather than one, creating a form of hybrid of the above approaches. They argued that the mixture approach seemed to minimize maximum losses and proved more robust than others, with the pure voting and bagging systems performing substantially worse than the other two. They concluded that creating committees would be a useful way to decrease noise and that in decision making, it might be useful to combine decisions rather than their forecasts.

An alternative to centralizing decisions is shown by Ontañón and Plaza [90]. The authors focused on committees of agents with learning capabilities where no agent is omniscient but has a local, limited and individual view of data. In their system, agents can solve any problem individually, however then try to create more accurate classifications through collaboration. Using a collaboration scheme based on symbolic justification of results, agent’s results are aggregated using a weighted voting scheme using confidence measures as weights. Different to previous research, the system does not contain a centralized method that has control over the entire training set. It could be argued that each agent is self-contained and only through their interaction does collectivism emerge, rather than enforcing unity through a centralized control
mechanism that has access to the internals of every agent, transforming them into transparent components of the overall system. They state that their method works well in different scenarios where normal voting tends to fail sometimes.

Schulenburg and Ross present an analysis of groups of agents trading in a stock market environment using historical data and a learning classifier system [109]. One of the key elements in their study was to focus on aspects of group decision-making rather than that of individual traders, based on the greater complexity of modeling a human individual with unique characteristics rather than as an abstract member of a group. Interestingly, one of their conclusions was that technical trading is a valid method to employ for trading in financial markets, but as with most things, can be less or more appropriate at times compared to non-technical trading. Furthermore, they suggest a form of ‘super trader’, who could base decisions on both technical and non-technical traders to overall hopefully perform better than either. This resembles group decision making, whereby a collective decision is taken from various ‘different’ types of agents that is then used to make overall trading decisions as one entity.

2.6 Summary

Numerous approaches have been taken to analysing financial markets for well over a century now, and more recently attempts have been made to simulate it with ever increasing realism to further the understanding of the underlying market dynamics. From this abundance of available research, various streams have emerged that focus
on the more sociological, psychological, financial, mathematical or simply behavioral aspects of people, society or the market itself. In this respect, a particular obsession has been the prediction of time series or more specifically in this instance, of market movements as represented by market indices. The prospect of automating the accumulation of wealth has lead to leading financial institutions pouring considerable sums of money into research and creating ever more sophisticated market models and algorithmic trading systems.

The above articles discussed a sample of the variety of applications and approaches taken to tackling the problem of successful trading in markets, employing artificial intelligence, agent based systems and other methodologies or combinations thereof. However, one notable area of promising research is the use of committees, also referred to as teams, schools or simply groups in wider literature, in the financial domain using agent technologies. Furthermore, there seems to be an absence of research drawing on well established theories of decision making in both individual and group contexts from the human domain and translating these into agent based systems. This therefore poses an interesting area of research, examining whether benefits observed in human-based group decision making could be transferred into agent based systems, in particular in the highly complex domain of financial markets.
CHAPTER 3

Experimentation System Outline

3.1 Introduction

To conduct the proposed investigations, agent populations emulating equity market traders were evolved using a variety of genetic algorithms. The agents analyse the market using a set of methodologies such as technical analysis to determine trading decisions.

Neither transaction costs nor interest on held capital were included, and the environment is assumed discrete and deterministic in a liquid market, meaning that an agent’s actions cannot affect prices. This is based on the assumption that the effect on an agent’s trading performance resulting from interest on capital holdings and that of transaction costs are relatively small compared to that resulting from trading activity, and therefore would have a negligible impact on performance. The assumption that an agent’s actions will not affect prices is a simplification of the model, as any
interaction with the market will always result in a change of volume and potentially the last traded price. However the traded volumes that can result from the assumed capital holdings in this simulation are negligible and would at best have a minor and temporary effect on the market.

3.2 Agent Design

Following on from the definition of an agent in Section 2.2, every agent represents an individual trader with a personal portfolio and capital holdings, collectively referred to as their total assets. With a fixed starting capital, at the end of every trading day each trader uses historical price data to make a decision on every security whether to buy, sell or hold. No limitations, apart from capital constraints, are set on the number of trades conducted.

Each agent follows a homogeneous decision process. This decision process is in each case however varied based on the agent’s genome, affecting both the method of analysis as well as parameters used throughout the decision process. This allows for the emergence of distinct phenotypes.

Parameters used in the agent’s analysis and decision process are determined by the agent’s genome, effectively determining its trading behavior. The genome for every agent consists of a string of integers taking values 1-10 and of length as determined by the methodology used by the agent, as detailed later in Section 3.6.

Depending on the investigation however, different methodologies were set for every
agent, which in turn fundamentally determined the decision process. Though necessary to allow for differences between the methodologies, changes were otherwise kept to a minimum to reduce the number of variable factors for later comparison.

3.3 Historical Market Data

The system uses historical financial data taken from the DAX-30, which is an index listing the top 30 capital weighted companies registered in the German market, with various weighting factors applied to each listed company to determine their impact on the Index. However the system does not include all securities from the DAX for its entire span, as due to changes in the constituents of the Index, daily closing price information was only available for 20 securities over the last few decades. Those included are listed in Table 3.1 using their Wertpapierkennummer (German security identification number).

The time range covered is specified in each experiment as appropriate. To provide an overview of data splits and their use in this series of experiments, Figure 3.1 represents a time line with the box segments indicating data sets used and Table 3.2 providing specific details. Sets A-F are used in Sections 6.2.1 and 6.2.2. Sets G-I are used only in Section 6.2.3 with set J used repeatedly in Sections 6.2.3, 6.3 and 6.4. Sets K and L are used in Chapters 4, 5 and 7 as well as Section 6.4.
Table 3.1: Securities used in the system

<table>
<thead>
<tr>
<th>Wertpapierkennnummer</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>840400</td>
<td>ALLIANZ AG VNA O.N.</td>
</tr>
<tr>
<td>760080</td>
<td>ALTANA AG O.N.</td>
</tr>
<tr>
<td>515100</td>
<td>BASF AG O.N.</td>
</tr>
<tr>
<td>575200</td>
<td>BAYER AG O.N.</td>
</tr>
<tr>
<td>802200</td>
<td>BAY.HYPO-VEREINSBK.O.N.</td>
</tr>
<tr>
<td>519000</td>
<td>BAY. MOTOREN WERKE AG ST</td>
</tr>
<tr>
<td>803200</td>
<td>COMMERZBANK AG O.N.</td>
</tr>
<tr>
<td>543900</td>
<td>CONTINENTAL AG O.N.</td>
</tr>
<tr>
<td>514000</td>
<td>DEUTSCHE BANK AG NA O.N.</td>
</tr>
<tr>
<td>761440</td>
<td>E.ON AG O.N.</td>
</tr>
<tr>
<td>648300</td>
<td>LINDE AG O.N.</td>
</tr>
<tr>
<td>823212</td>
<td>DT. LUFTHANSA AG VNA O.N.</td>
</tr>
<tr>
<td>593700</td>
<td>MAN AG ST O.N.</td>
</tr>
<tr>
<td>843002</td>
<td>MUENCH.RUECKVERS.VNA O.N.</td>
</tr>
<tr>
<td>703712</td>
<td>RWE AG ST O.N.</td>
</tr>
<tr>
<td>717200</td>
<td>SCHERING AG O.N.</td>
</tr>
<tr>
<td>723610</td>
<td>SIEMENS AG NA O.N.</td>
</tr>
<tr>
<td>750000</td>
<td>THYSSENKRUPP AG O.N.</td>
</tr>
<tr>
<td>TUAG00</td>
<td>TUI AG NA O.N.</td>
</tr>
<tr>
<td>766400</td>
<td>VOLKSWAGEN AG ST O.N.</td>
</tr>
</tbody>
</table>

Figure 3.1: Data splits used

3.4 Evolutionary Approaches

Following from an agent based approach, genetic algorithms lend themselves well to evolve agent populations [12] and are able to evolve very robust solutions [10]. As
3. EXPERIMENTATION SYSTEM OUTLINE

Table 3.2: Data splits used

<table>
<thead>
<tr>
<th>Data set</th>
<th>Time range</th>
<th>Trading days covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>01.01.88 to 29.12.89</td>
<td>521</td>
</tr>
<tr>
<td>B</td>
<td>01.01.85 to 29.12.89</td>
<td>1304</td>
</tr>
<tr>
<td>C</td>
<td>01.01.80 to 29.12.89</td>
<td>2609</td>
</tr>
<tr>
<td>D</td>
<td>01.01.90 to 31.12.91</td>
<td>522</td>
</tr>
<tr>
<td>E</td>
<td>01.01.90 to 30.12.94</td>
<td>1305</td>
</tr>
<tr>
<td>F</td>
<td>01.01.90 to 31.12.99</td>
<td>2610</td>
</tr>
<tr>
<td>G</td>
<td>01.01.1980 to 28.12.1984</td>
<td>1305</td>
</tr>
<tr>
<td>H</td>
<td>01.01.1985 to 29.12.1989</td>
<td>1304</td>
</tr>
<tr>
<td>I</td>
<td>01.01.1990 to 30.12.1994</td>
<td>1305</td>
</tr>
<tr>
<td>J</td>
<td>01.01.1997 to 31.12.2002</td>
<td>1826</td>
</tr>
<tr>
<td>K</td>
<td>01.01.1990 to 29.12.1995</td>
<td>1465</td>
</tr>
<tr>
<td>L</td>
<td>01.01.1996 to 31.12.1996</td>
<td>262</td>
</tr>
<tr>
<td>J</td>
<td>01.01.1997 to 31.12.2002</td>
<td>1826</td>
</tr>
</tbody>
</table>

Cawsey put it, good solutions or designs can evolve out of a population, by combining possible solutions to produce offspring solutions and removing the weaker [23]. Six separate genetic algorithms were implemented in this work. The algorithm originally introduced in [107] and later related works will be presented briefly and also form the benchmark for comparison. The five remaining algorithms and their implementations will on the other hand be described in depth in this section, outlining their key differences. Each is based on a combination of fitness assessment, followed by roulette or tournament selection and eventually by crossover and mutation operations. Some include ranked fitness and fitness scaling elements.
3.4.1 Introduction

As previously introduced, Allen and Karjalainen published results using a genetic algorithm to evolve trading strategies [4]. Using daily closing price information and at varying transaction cost levels, they found that their evolved rules did not consistently earn excess returns over a simple buy-and-hold strategy. The buy-and-hold strategy being a common reference benchmark, where the entire starting capital is evenly invested in the available assets for the entire time period under consideration. They stated that there had been little formal analysis in financial economics of genetic algorithms and that the purpose of their paper was to demonstrate how genetic algorithms could be used to find technical trading rules. Further work was performed by Potvin et al, who used genetic programming to again evolve trees of technical trading rules [98]. They found that their trading rules performed well during stable or falling markets, while in appreciating markets the buy-and-hold strategy fared better. In general, genetic algorithms have been used in a wide variety of financial applications by many authors [13, 63, 88, 97].

3.4.2 Original Genetic Algorithm

For selection, elitism [48] is used, whereby a portion of the most successful agents carries forward unaltered every generation. Immigration [17] was also employed, where a portion of the worst performers are removed and replaced by a new randomly generated group of agents immigrating into the system, constantly introducing new
genetic material. This facilitates greater coverage of the search space, while also avoiding premature convergence and non-exclusion of other possible solutions not present in the original base population’s gene pool. Each agent in the mediocre population, those not removed or part of the elite, randomly selects another agent from the mediocre and elite population and uses two-point crossover to create an offspring. Two-point crossover having been chosen as it facilitates greater exploration of the search space than the popular one-point crossover, as well as reducing the premature convergence on one particular solution. This offspring then replaces the parent from the mediocre population [16, 120, 138]. Two randomly selected agents of the same decision type can mate, with a random part of the first agent’s genome being replaced by the equivalent section from the second agents genome, forming a new genome combination. The only restriction in this process is that there exists a 25% chance of an agent mating with an agent of a different decision type. This may cause a significantly different phenotype to emerge, as for instance weighting genes will not have been relevant to DT1 or DT2 previously. If an agent of DT1 should mate with an agent of DT3, producing an offspring of DT3 with the weighting related genes $G_4$-$G_{11}$ identical to its DT1 parent for example, $G_4$-$G_{11}$ will thus far have had no influence on the agents phenotype and therefore not been optimized, and could have a severely negative effect on the agents performance. Offspring are included in the new population regardless of their fitness. A general representation of the algorithm can be seen in Figure 3.2.
3. Experimentation System Outline

3.4.3 Simple Genetic Algorithm

The simple genetic algorithm (sGA), as the name implies, aims to represent a basic implementation of a Holland-type genetic algorithm. Agents are first ranked based on the fitness function, with each agent allocated a fitness value between 0 and a 100. This fitness value is determined by dividing the value returned from their fitness function by the cumulative fitness value of all agents. For example, in a simulation with 3 agents and respective fitness function values of 4, 8 and 12, the first agent’s
fitness value would be 4 divided by the sum of 4+8+12 (i.e. \( 4 \div 24 = \frac{1}{6} \)). This would then be converted into a proportionate percentage of approximately 17. This fitness value then feeds into the roulette selection, where a random number between 0 and a 100 is generated. In this scenario, the agent with the greatest fitness value will therefore have the highest probability of being selected. Continuing from the example above, the third agent would thus have a selection probability of 50%. Once a population of agents has been selected, each agent mates with another randomly selected agent using two-point crossover, with the offspring replacing the non-randomly selected parent in the next population pool. Finally, the offspring population is mutated at a 0.001 probability prior to starting the next simulation run, following some preliminary exploration of varying mutation rates.

### 3.4.4 Tournament Genetic Algorithm

The tournament genetic algorithm (tGA) is based on the same steps as the sGA, except that it uses tournament rather than roulette selection. Therefore after assigning fitness values to each agent, using binary selection two agents are randomly selected from the population, the fitter of the two being added to the subsequent population for crossover and later mutation. When both agents have an identical fitness level in selection, either is chosen randomly.
3. EXPERIMENTATION SYSTEM OUTLINE

3.4.5 Ranked Simple Genetic Algorithm

The ranked simple genetic algorithm (rsGA) forms an extension of the sGA. The ranking function first ranks all agents based on the value returned by the fitness function, then replacing the absolute value by their rank relative to each other. This ranking takes a value between 1 and 100, based on the sum of ranks of all agents. For instance, agents with the highest fitness returned by the fitness function will be ranked 1, 2, 3 etc. respectively. The ranked fitness assigned to them will replace their absolute fitness function value with the best agent being assigned 100, the second best 99 and so on. This value is then converted to a relative percentage based on the sum of all rank fitness values, which for the best agent would be $\frac{100}{\sum_{n=1}^{100} n} = \frac{2}{101}$, for the second best $\frac{99}{\sum_{n=1}^{99} n} = \frac{99}{5050}$ and so on. Changing the fitness value to a rank based fitness value affects the probabilities in the roulette selection, effectively disregarding the absolute differences in fitness and replacing it with a relative measure based on their ranking. The next steps in the rsGA are identical to those of the sGA.

3.4.6 Ranked Simple Genetic Algorithm with Fitness Scaling

The ranked simple genetic algorithm with fitness scaling (rsGAFs) further extends the rsGA by implementing a fitness scaling function based on Goldberg’s work [48]. Following the modified fitness based on agents’ relative ranking, the fitness scaling function is implemented prior to the roulette wheel selection to encourage the persistence of greater diversity in the population. In fitness scaling, an agent’s similarity in
terms of its genome is assessed, with the fitness of an agent decreasing as similarity increases. In our simulation, an average genome is first derived by taking the mean of every single gene in the agent population to which every agent is then compared in a correlation analysis. Agents are then subsequently given a ranking based on their correlation to that average genome, with the lowest correlation therefore being ranked first for instance. Both rankings for fitness and fitness scaling are then added together to give their final fitness. For example, the best performing agent will have been ranked first by fitness, may hypothetically however also have displayed the greatest genome similarity and have been ranked lowest in that respect. This will have assigned it a final fitness value identical to the worst performing agent that also showed the least similarity to the average genome for example. In other words, agents with non-uniform genomes that perform well will have the highest probability of being selected for reproduction. As an example, an agent with the lowest similarity but also worst performance will have been assigned ranks of 100 and 1 respectively, giving a final value of \(\sum_{n=1}^{100+1} 2n = \frac{101}{100} = \frac{1}{100} = 1\%\).

3.4.7 Ranked Tournament Genetic Algorithm with Fitness Scaling

The ranked tournament genetic algorithm with fitness scaling (rtGAs) represents the final variation of the algorithms implemented, forming an extension to the tGA. It combines the fitness ranking with tournament selection, while simultaneously incorporating the fitness scaling function last introduced in the rsGAs algorithm.
3.5 Fitness Measures

Several fitness measures have further been implemented to investigate their impact on evolving a system capable of producing consistent returns. To allow for comparison, the area under their total assets graph will be used as the performance measure when testing for all methods. This measure will be introduced as fitness type 1.

3.5.1 Fitness Type 1 - Area under Performance Graph

In wider literature, performance tends to generally be measured as an agent’s capital and value of all holdings at the end of the trading period relative to its starting capital. However, as this biases results based on the cut-off date used, a more overall and therefore more accurate picture of performance throughout the entire testing period can be obtained by using the area under an agent’s total asset graph. When using this fitness type, hereafter abbreviated to FT1, the fitness function will refer to the area under an agent’s total asset graph for the trading period being assessed. The importance of this becomes apparent when an agent’s or group of agents’ performance is consistently better than that of its competitors, but at one point suffers unusually large losses. If performance is measured at this point, that particular agent will then appear to be a below average trader. Hence, by using the area under an agent’s total asset graph for comparison, this eliminates the possibility of misinterpreting overall performance by measuring it at only one instance in time. Mathematically this can be presented as:
where \( t \) represents the total number of trading days and \( totalAssets(n) \) representing the total assets on trading day \( n \).

### 3.5.2 Fitness Type 2 - Ratio of Daily Up or Down Changes in Total Assets

The fitness function for fitness type 2 (FT2) is based on the daily change in total assets held by an agent. It determines the total number of times where the value of an agent’s total assets changed upwards and downwards from the previous trading day. A ratio is taken between the two values with a large value reflecting a portfolio that predominately experienced increases in value on a day to day basis, while a ratio below 1 would indicate more frequent decreases in value. Most importantly this does not take account of the magnitude in daily changes, merely whether an increase or decrease was experienced on a day to day basis. On the other hand however, a situation may arise where an agent with a high ratio may not actually have performed very well. For example, over a time span of 20 days the agent may have initially experienced 15 days where the portfolio appreciated by 0.1% per day and 5 consecutive days with losses of 0.5% each day at the end of the trading period. Therefore overall it will have made a significant loss while its fitness ratio would be \( \frac{15}{5} = 3 \), indicating that most of the time that agent’s assets increased in value on a day to day basis. This can be shown as:

\[
fitnessFT1(t) = \sum_{n=1}^{n=t} totalAssets(n)
\]
$$fitness_{FT2}(t) = \frac{\sum_{n=1}^{t} ups(n)}{\sum_{n=1}^{t} downs(n)}$$  \hspace{1cm} (3.2)$$

where $ups(n)$ refers to the number of times a gain was made, while $downs(n)$ represents the number of decreases.

### 3.5.3 Fitness Type 3 - Profit

The third fitness function for fitness type 3 (FT3) is among the most commonly used to determine trading performance, as it uses the profit made over a time period. Profit will therefore refer to the total assets held at the end of trading relative to an agent’s starting capital. As discussed for FT1, using profit as the fitness function will focus solely on their gains or losses at the end of the trading period. For example, an agent that has performed disastrously for the majority of the time may manage to triple its assets over a short time period and outperform another agent who has been producing consistent returns, albeit of a lesser magnitude. In another similar scenario using two agents, agent A could make a profit of 30% on its one and only trade early in the trading period, while agent B makes a profit of 30% on its only trade very near the end of the trading period. Using FT1 agent A would clearly appear superior having a much larger area under its total assets curve, while using FT3 both would be recorded as having generated a profit of 30% over the entire trading period. Though it could be argued that generating a profit early in the trading period is more desirable as reflected in FT1, FT3 disregards this. This can be shown as:
3.5.4 Fitness Type 4 - Ratio of Daily Up or Down Percentage Changes in Total Assets

Similar to FT2, fitness type 4 (FT4) uses the daily change in daily assets to determine fitness. However, it uses the magnitude of the change as a percentage relative to the preceding day rather than simply recording an increase or decrease in total assets as in FT2. Taking the sum of all daily changes at the end of the trading period, a positive sum would indicate that on average the agent generated a profit. If the sum of daily changes returned a negative value, this would indicate that overall the agent was loss making. Using the same example as for FT2, in this scenario the equation to calculate the fitness would be \((15 \cdot 0.1) - (5 \cdot 0.5) = 1.5 - 2.5 = -1\). Clearly the fitness for this agent using FT4 when compared to FT2 is negative, suggesting it did not perform well. Mathematically:

\[
\text{fitnessFT4}(t) = \sum_{n=2}^{n=t} \frac{\text{totalAssets}(n)}{\text{totalAssets}(n-1)}
\]  

(3.4)

3.5.5 Fitness Type 5 - Ratio of Profit versus Loss Trades

Moving away from using total assets in determining fitness, fitness type 5 (FT5) uses the ratio of profit making trades versus loss making trades. In other words, it
compares how many completed trades over the entire trading span made a profit or loss when sold. For example, an agent makes five trades over the trading duration, with the same security at different points in time or otherwise, 3 of which are sold at a profit and 2 of which are sold at a loss. This would give it a fitness value of $\frac{3}{2} = 1.5$ using FT5, with larger values suggesting a better trading strategy as more trades ended in a profit rather than a loss. Again as in FT2, this does not take into account the magnitude of the profits or losses incurred. If no losses at all were incurred over time $t$, the value of 0.5 was assumed as the denominator. Expressed mathematically:

$$\text{fitnessFT5}(t) = \frac{\sum \text{profitTrades}(t)}{\sum \text{lossTrades}(t)} \quad (3.5)$$

where $\text{profitTrades}(t)$ represents the number of trades that made a profit over the time period $t$, and $\text{lossTrade}(t)$ represents the number of loss making trades over the time period $t$. 
3.6 Methodologies

3.6.1 Introduction

Technical Analysis is trend based, providing an indication as to the future movement of the time series under consideration. Research into the success of technical trading suggests that it can act as a good indicator for making trading decisions [75, 56, 2]. When compared to the Capital Asset Pricing Model, a main distinguishing feature is the introduction of risk, producing a more comprehensive analysis of the situation and a comparative ranking between alternatives. This section will describe the methodologies employed in analysis and the associated decision making process for each agent.

3.6.2 Technical Analysis

A significant number of papers employ technical analysis in their strategy or portfolio optimization simulations, as discussed in Section 2.4. Though there exists a theoretical divide on the feasibility of technical analysis, its wide application and success of past systems that have employed technical analysis appear to justify its use as a tool in further research.

*Technical Indicators*

Technical indicators are tools used in the technical analysis of financial markets, exploiting the existence of trends to determine potential buy, sell or hold conditions.
Indicators are mathematical formulae, commonly based on closing price or volume data, with price information being used exclusively in this system.

Indicators used here are:

i. Simple Moving Average (MA)

ii. Relative Strength Index (RSI)

iii. Price-Rate-of-Change (ROC)

iv. Stochastic Oscillator (SO)

v. Moving Average Convergence Divergence (MACD)

vi. Bollinger Bands (BB)

Every agent uses technical indicators to generate trading signals for every security that is assessed. Depending on the signals returned, an overall decision of buy, sell or hold is made for a security depending on the agent’s decision type. Securities that are sold are converted into capital at their current closing price. The amount of capital invested is determined by the genome, with it equally allocated between each security flagged for acquisition. The genome for technical indicators is as detailed as in Table 3.3. Particular mention needs to be made of genes 1, 2, 21 and 28, as these deviate from the standard cardinality of 10. Gene 1, defining decision types, has a cardinality of 4, where gene 2, the risk averseness factor, has a cardinality of 2 while
3. EXPERIMENTATION SYSTEM OUTLINE

genomes 21 and 28, the Stochastic Oscillator variable D and Bollinger Band deviation
variable respectively, have a a cardinality of 5.

Table 3.3: Gene descriptions

<table>
<thead>
<tr>
<th>Gene</th>
<th>Value range</th>
<th>Description / Function</th>
<th>Local approx.</th>
<th>Lit. approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>1-4</td>
<td>Decision type</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_2$</td>
<td>1-2</td>
<td>Risk averseness factor</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_3$</td>
<td>1-10</td>
<td>Capital investment proportion</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_4$</td>
<td>1-10</td>
<td>Moving Average weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_5$</td>
<td>1-10</td>
<td>RSI weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_6$</td>
<td>1-10</td>
<td>Short-term ROC weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_7$</td>
<td>1-10</td>
<td>Long-term Price ROC weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_8$</td>
<td>1-10</td>
<td>SO interpretation 1 weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_9$</td>
<td>1-10</td>
<td>SO interpretation 2 weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_{10}$</td>
<td>1-10</td>
<td>MACD weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_{11}$</td>
<td>1-10</td>
<td>BB weight</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$G_{12}$</td>
<td>1-10</td>
<td>MA short-term value</td>
<td>22</td>
<td>14-25</td>
</tr>
<tr>
<td>$G_{13}$</td>
<td>1-10</td>
<td>MA long-term value</td>
<td>77.5</td>
<td>50-100</td>
</tr>
<tr>
<td>$G_{14}$</td>
<td>1-10</td>
<td>RSI time period</td>
<td>13.75</td>
<td>14</td>
</tr>
<tr>
<td>$G_{15}$</td>
<td>1-10</td>
<td>RSI buy threshold</td>
<td>22.5</td>
<td>30</td>
</tr>
<tr>
<td>$G_{16}$</td>
<td>1-10</td>
<td>RSI sell threshold</td>
<td>72</td>
<td>70</td>
</tr>
<tr>
<td>$G_{17}$</td>
<td>1-10</td>
<td>ROC level</td>
<td>5.5</td>
<td>-</td>
</tr>
<tr>
<td>$G_{18}$</td>
<td>1-10</td>
<td>ROC short-term value</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>$G_{19}$</td>
<td>1-10</td>
<td>ROC long-term value</td>
<td>22</td>
<td>25</td>
</tr>
<tr>
<td>$G_{20}$</td>
<td>1-10</td>
<td>SO K variable value</td>
<td>8.25</td>
<td>-</td>
</tr>
<tr>
<td>$G_{21}$</td>
<td>1-5</td>
<td>SO D variable value</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>$G_{22}$</td>
<td>1-10</td>
<td>SO buy threshold</td>
<td>19.25</td>
<td>20</td>
</tr>
<tr>
<td>$G_{23}$</td>
<td>1-10</td>
<td>SO sell threshold</td>
<td>72</td>
<td>80</td>
</tr>
<tr>
<td>$G_{24}$</td>
<td>1-10</td>
<td>MACD short-term value</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>$G_{25}$</td>
<td>1-10</td>
<td>MACD long-term value</td>
<td>24.75</td>
<td>26</td>
</tr>
<tr>
<td>$G_{26}$</td>
<td>1-10</td>
<td>MACD signal line</td>
<td>8.25</td>
<td>9</td>
</tr>
<tr>
<td>$G_{27}$</td>
<td>1-10</td>
<td>BB time period value</td>
<td>19.25</td>
<td>20</td>
</tr>
<tr>
<td>$G_{28}$</td>
<td>1-5</td>
<td>BB deviations number</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

The following is a brief description of the indicators employed, and how an agent’s
genome is used to individualize calculations. The following descriptions are primarily
based on a summary presented by Achelis [2], and also include the translation of a number of gene values. When an agent is initialised, some of its gene values would not be appropriate for direct use in the system and need to be modified. For instance, though $G_{17}$ can be used directly in the technical indicator without translation, this is not possible for $G_{13}$. As an informal rule, the translation was based on achieving a representative average value approximate to the range suggested in literature. These two measurements are further listed in Table 3.3 as “Local approx.” and “Lit. approx.” respectively for convenience.

A MA shows the average value of a security’s price over time. The short-term and long-term moving average values are calculated as:

$$\text{MA}(t, N) = \frac{\sum_{i=t-N}^{t} \text{price}_i}{N}$$

(3.6)

where $N = 4G_{12}$ for the short-term and $N = 5G_{13} + 50$ for the long term, with $t$ being the current trading day.

The RSI is a price-following oscillator that measures the magnitude of gains and losses of a single security over a specific time period to determine the current trend. It is calculated as:

$$\text{RSI}(t, N) = \frac{100}{1 + \frac{\text{avgDowns}(t, N)}{\text{avgUps}(t, N)}}$$

(3.7)

where $N = 2.5G_{14}$ and $\text{avgUps}(t, N)/\text{avgDowns}(t, N)$ is the average of the increase/decrease in price noted for every day it closed higher/lower over the last $N$ days from day $t$. 

3. EXPERIMENTATION SYSTEM OUTLINE

The ROC indicator is based on the assumption of cyclical price movements, and considers the relative change of prices over time to indicate trends. It is calculated by:

\[
ROC(t, N) = \frac{price_t - price_{t-N}}{price_t}
\] (3.8)

where \( N = 2G_{18} \) for the short-term and \( N = 4G_{19} \) in the long-term.

The SO compares a security’s price relative to its price range over a given time period, using two parameters commonly termed as \( K \) and \( D \). \( K \) is calculated by:

\[
K(t, N) = 100\left( \frac{price_t - \text{lowestClose}(t, N)}{\text{highestClose}(t, N) - \text{lowestClose}(t, N)} \right)
\] (3.9)

where \( N = 1.5G_{20} \) and where \( D(t, N) \) is a moving average of \( K \) over \( G_{21} \) days.

The MACD is a trend following momentum indicator that shows the relationship between two moving averages of prices. The MACD is calculated:

\[
\text{exMA}(t, N) = \frac{2price_t + \text{prevMA}(N - 1)}{N + 1}
\] (3.10)

\[
\text{MACD}(t, N, O) = \text{exMA}(t, N) - \text{exMA}(t, O)
\] (3.11)

where \( N = 2G_{24} \), \( O = 4.5G_{25} \) and prevMA is the previous exponential moving average apart from the first instance, where a simple moving average is used.

Lastly, BBs are generally used to provide a form of guideline, indicating possible trend reversals. The upper and lower bands are calculated as:

\[
\text{BBstdDev}(t, N) = G_{28} \sum_{i=t-N}^{t} (price_i - \text{MA}(i, N))^2
\] (3.12)

\[
\text{upperBand}(t, N) = \text{MA}(t, N) + \text{BBstdDev}(t, N)
\] (3.13)
lowerBand\((t, N)\) = \(MA(t, N) - BBstdDev(t, N)\) \hspace{1cm} (3.14)

where \(N = 3.5G_{27}\).

Decision Process and Decision Types

There are essentially three steps that every agent using technical analysis follows to determine whether or not to add a particular security to its acquisition or sale list. To analyse a security it first performs the calculations described above, each using historical closing price information. Depending on the indicator, one or more values are returned which then need to be interpreted for the calculated values to gain meaning. Based on this interpretation, every agent then compiles a list of buy and sell signals for every security. The final step is generating an overall buy, sell or hold decision for each security, based on the agent’s decision type. The following descriptions provide the decision process and interpretation for each indicator, with the interpretation again based on established practice and presented in [2]. For notational convenience, the parameter \(t\) was dropped in the pseudo code following.

If the MA over the short-term is larger than over the long-term, it indicates an upward trend and a buy signal would be generated. If the MA over the short-term is smaller than the long-term, a downward trend is indicated and a sell signal would be generated. Additionally, the short-term MA can be compared to the current price of the security, which if greater, would indicate a downward trend and hence a sell signal should be generated. In this implementation, if the agent is risk averse, as
determined by $G_2$, it bases its interpretation on a logical **AND** between those two interpretations and is therefore more reluctant to generate a buy signal. On the other hand, if the agent is risk taking, a logical **OR** is used and either interpretation suggesting a buy would suffice for the agent to consider this a buy signal. In pseudo-code:

\[
\begin{align*}
\text{BOOLEAN A} &= \text{MA}(4G_{12}) < \text{current price} \\
\text{BOOLEAN B} &= \text{MA}(5G_{13}+50) < \text{MA}(4G_{12}) \\
\text{BOOLEAN C} &= (G_2 == 1) \\
\text{IF ( A AND B AND C ) OR (NOT C AND ( A OR B ))} \\
& \quad \text{Action: buy} \\
\text{ELSE} \\
& \quad \text{Action: sell} \\
\text{ENDIF}
\end{align*}
\]

For the RSI, if the calculated value lies above or below the sell or buy threshold respectively, the appropriate signal will be generated.

\[
\begin{align*}
\text{IF RSI}(2.5G_{14}) &\geq 4G_{16}+50 \\
& \quad \text{Action: sell} \\
\text{ELSEIF RSI}(2.5G_{14}) &\leq 5G_{15} \\
& \quad \text{Action: buy} \\
\text{ELSE}
\end{align*}
\]
The ROC is repeated for both long- and short-term analysis. If the calculated value lies below the negative threshold value it indicates a buy, while a calculated value above the positive threshold value indicates a sell in both repeats. The following example is for the short-term, with ROC(2G\textsubscript{18}) being replaced by ROC(4G\textsubscript{19}) for the long-term.

\[
\text{IF } \text{ROC}(2G_{18}) < -G_{17} \\
\text{Action: buy}
\]

\[
\text{ELSEIF } \text{ROC}(2G_{18}) > G_{17} \\
\text{Action: sell}
\]

For the SO multiple interpretations are possible, though the following two are used in this instance. First, it can be considered a buy signal if the $K$ value is larger than the $D$ value or a sell signal if $D$ is larger than $K$. Second, threshold values can be added for both $K$ and $D$. In that case, if $K$ and/or $D$ is smaller than the buy threshold a buy signal is generated, or vice versa for a sell signal, if $K$ and/or $D$ is larger than the sell threshold a sell signal is generated.

\[
\text{IF } K(1.5G_{20}) > D(G_{21})
\]
Action: buy

ELSE

Action: sell

ENDIF

BOOLEANA = K(1.5G) ≥ 3.5G
BOOLEANB = D(G) < 3.5G
BOOLEANC = K(1.5G) > (4G)+50
BOOLEAND = D(G) > (4G)+50
IF (G AND A AND B) OR (NOT G AND (A OR B))
Action: buy
ELSEIF (G AND C AND D) OR (NOT G AND (C OR D))
Action: sell
ENDIF

The MACD compares its calculated value to a moving average of itself over a time period, whereby a buy signal is generated if the moving average is smaller, and a sell signal if the moving average is larger.

IF MACD(2G, 4.5G) > MA(1.5G, MACD(2G, 4.5G))
Action: buy
ELSE
Action: sell
ENDIF

BBs indicate that when the current price breaks through the lower bollinger band it is considered a buy signal, while if it breaks through the upper band it is considered a sell signal.

IF lowerBand(3.5G27) >= current price
Action: buy
ELSEIF upperBand(3.5G27) <= current price
Action: sell
ENDIF

Thus the technical indicators generate up to eight buy/sell/hold signals. In an attempt to allow for different approaches that exist among real traders to selecting securities for purchase or sale, four methods of combining the eight individual signals into an overall decision were implemented.

i. Decision type 1 (DT1) performs a simple comparison between the number of buy and sell signals, making the appropriate decision if one is greater than the other for any particular security. For instance, out of the 8 possible signals used here for a security, if 3 are buy and 2 are sell, an agent of this type would decide to purchase this security.

\[ \text{numberOfBuys} > \text{numberOfSells} \equiv \text{Buy} \quad (3.15) \]
3. EXPERIMENTATION SYSTEM OUTLINE

\[ \text{numberOfBuys} < \text{numberOfSells} \equiv \text{Sell} \] (3.16)

ii. Decision type 2 (DT2) follows the same principle as DT1. However it also stipulates that for a buy or sell decision to be made, at least half of all signals must be in favor. In this instance for example, out of 8 possible signals, if only 3 are buy signals even though no sell signals exist, no decision will be taken as it failed to reach the minimum buy threshold.

\[ \text{numberOfBuys} > \frac{\text{totalSignals}}{2} \equiv \text{Buy} \] (3.17)

\[ \text{numberOfSells} > \frac{\text{totalSignals}}{2} \equiv \text{Sell} \] (3.18)

iii. Decision type 3 (DT3) sums the indicators by taking buy signals as +1 and sell signals as -1, while including a weighting process on each signal \((G_4 \text{ to } G_{11})\), increasing or decreasing its impact on the final sum. Therefore a positive sum would translate into an overall buy decision, while a negative sum into an overall sell decision.

\[ \text{signal}_i \cdot \text{weight}_i > 0 \equiv \text{Buy} \] (3.19)

\[ \text{signal}_i \cdot \text{weight}_i < 0 \equiv \text{Sell} \] (3.20)

iv. Decision type 4 (DT4) follows the same principle as DT3, except for adjusting the final sum to create a threshold value of ±10 which it needs to exceed prior to resulting in an overall buy or sell decision. For example, though decision type 3 would generate a buy decision for a value of +5, for decision type 4 this
would produce a neutral hold decision.

\[
N = \text{totalSignals} \\
\sum_{i=1}^{N} \text{signal}_i \cdot \text{weight}_i > 10 \equiv \text{Buy} \tag{3.21}
\]

\[
N = \text{totalSignals} \\
\sum_{i=1}^{N} \text{signal}_i \cdot \text{weight}_i < -10 \equiv \text{Sell} \tag{3.22}
\]

The entire process from technical analysis to the purchase and sale of securities is summarized in Figure 3.3.
Figure 3.3: Decision process from technical analysis to acquisition and sale of securities
3.6.3 Capital Asset Pricing Model

The Capital Asset Pricing Model is based on Markowitz’s portfolio theory [78] where a portfolio’s overall exposure to risk is reduced through a balanced selection of assets. Essentially, the Capital Asset Pricing Model determines the attractiveness of a particular security in form of a numerical value, which then allows selection and comparison between possible investments. The higher the value, the more desirable the investment appears. The attractiveness of a security is its expected rate of return, an estimate as to what an investment in that security would yield within the specified time frame. In an agent population that uses the Capital Asset Pricing Model, both security selection and capital allocation between the securities selected for acquisition are determined by the model. A significantly reduced number of parameters are required in its calculation compared to the Technical Analysis model, resulting in a smaller genome length. Specifically, three genes were used, with both $G_1$ and $G_3$ taking integer values of 1-10 inclusive and $G_2$ representing a boolean value. $G_1$ and $G_3$ are both further translated by multiplying with a constant value of 10, where $G_1$ is to represent the number of trading days taken under consideration in the calculation and $G_3$ the proportion of available capital to be invested, similar to the method used in the Technical Analysis based approach. The multiplier of 10 extending the time period considered to a more representative range. $G_2$ represents the choice between using ex-ante or a moving average for the prediction of the equity risk premium. $G_4$ and $G_5$ represent the respective minimum sell and buy difference between the ex-
pected rate of return and the risk-free interest alternative when determining whether
to add a security to the buy or sell list. A larger value for $G_4$ or $G_5$ would therefore
increase the necessary estimated return before a purchase or sale is made. Finally,
the sum of $G_6$ and $G_7$ is used to determine the maximum number of securities that
an agent will purchase, starting with the securities that have the largest estimated
return. This is summarized in Table 3.4.

**Table 3.4: Capital Asset Pricing Model gene descriptions**

<table>
<thead>
<tr>
<th>Gene</th>
<th>Value range</th>
<th>Description / Function</th>
<th>Local approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>1-10</td>
<td>Time frame</td>
<td>55</td>
</tr>
<tr>
<td>$G_2$</td>
<td>1-2</td>
<td>ERP prediction choice between ex-ante or MA</td>
<td>-</td>
</tr>
<tr>
<td>$G_3$</td>
<td>1-10</td>
<td>Capital investment proportion</td>
<td>55</td>
</tr>
<tr>
<td>$G_4$</td>
<td>0-9</td>
<td>Minimum sell difference</td>
<td>4.5</td>
</tr>
<tr>
<td>$G_5$</td>
<td>0-9</td>
<td>Minimum buy difference</td>
<td>4.5</td>
</tr>
<tr>
<td>$G_6$</td>
<td>1-10</td>
<td>Securities to invest in (1)</td>
<td>5.5</td>
</tr>
<tr>
<td>$G_7$</td>
<td>1-10</td>
<td>Securities to invest in (2)</td>
<td>5.5</td>
</tr>
</tbody>
</table>

The method employed in this investigation to calculate the expected rate of return
of the available securities is shown by the following equation:

$$E(r) = R(f) + \beta(R(m) - R(f))$$  \hspace{1cm} (3.23)

where $E(r)$ is the expected return of the security under consideration, $R(f)$ is
the risk free rate of return such as a fixed-yield bond, $\beta$ represents the measure of
risk and $R(m)$ is the return possible in the current market. In the Capital Asset
Pricing Model, there are two methods of calculating $R(m)$, the estimated return of
the market in percent over the time period determined by $G_1$. The first is to use
historical data to calculate the average change over the period, while the second is to try to predict the future price by using the historical moving average. In this implementation the agent chooses between either method based on $G_2$. Once the expected return for all securities has been calculated, all securities with a negative value are dismissed for purchase and placed on the sales list. The available capital is then divided among those on the purchase list based on their relative expected rate of return. For example, if two securities A and B are to be purchased with expected returns of 5 and 15 respectively, 25% of the proportion of available capital to be invested based on $G_3$ would be invested in A and 75% would be invested in B.

3.6.4 Hybrid of Technical Analysis and Capital Asset Pricing Model

A hybrid model of both Technical Analysis and the Capital Asset Pricing Model was further implemented, where Technical Analysis determined the selection of securities, while the Capital Asset Pricing Model determined the allocation of capital among those selected. Additionally, securities selected through Technical Analysis that were predicted a negative return by the Capital Asset Pricing Model were dismissed in a form of secondary selection process. To accommodate the increased complexity of the hybrid approach, the genome increased in size incorporating elements from both approaches, resulting in a length of 30. It is important to note however that as $G_3$ exists in both genomes and is identical in function, it is not duplicated. The technical indicators, agent decision types and all other elements described in Section 3.6.2 and
Section 3.6.3 apply in the hybrid model. The genome is summarized in Table 3.5.

**Table 3.5**: Technical Analysis and Capital Asset Pricing Model hybrid gene descriptions

<table>
<thead>
<tr>
<th>Gene</th>
<th>Value range</th>
<th>Description / Function</th>
<th>Local approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>1-10</td>
<td>Time frame</td>
<td>55</td>
</tr>
<tr>
<td>$G_2$</td>
<td>1-2</td>
<td>ERP prediction choice between ex-ante or MA</td>
<td>-</td>
</tr>
<tr>
<td>$G_3$</td>
<td>1-10</td>
<td>Capital investment proportion</td>
<td>55</td>
</tr>
<tr>
<td>$G_4$</td>
<td>1-4</td>
<td>Decision type</td>
<td>-</td>
</tr>
<tr>
<td>$G_5$</td>
<td>1-2</td>
<td>Risk averseness factor</td>
<td>-</td>
</tr>
<tr>
<td>$G_6$</td>
<td>1-10</td>
<td>Moving Average weight</td>
<td>-</td>
</tr>
<tr>
<td>$G_7$</td>
<td>1-10</td>
<td>RSI weight</td>
<td>-</td>
</tr>
<tr>
<td>$G_8$</td>
<td>1-10</td>
<td>Short-term ROC weight</td>
<td></td>
</tr>
<tr>
<td>$G_9$</td>
<td>1-10</td>
<td>Long-term Price ROC weight</td>
<td>-</td>
</tr>
<tr>
<td>$G_{10}$</td>
<td>1-10</td>
<td>SO interpretation 1 weight</td>
<td>-</td>
</tr>
<tr>
<td>$G_{11}$</td>
<td>1-10</td>
<td>SO interpretation 2 weight</td>
<td>-</td>
</tr>
<tr>
<td>$G_{12}$</td>
<td>1-10</td>
<td>MACD weight</td>
<td>-</td>
</tr>
<tr>
<td>$G_{13}$</td>
<td>1-10</td>
<td>BB weight</td>
<td>-</td>
</tr>
<tr>
<td>$G_{14}$</td>
<td>1-10</td>
<td>MA short-term value</td>
<td>22</td>
</tr>
<tr>
<td>$G_{15}$</td>
<td>1-10</td>
<td>MA long-term value</td>
<td>77.5</td>
</tr>
<tr>
<td>$G_{16}$</td>
<td>1-10</td>
<td>RSI time period</td>
<td>13.75</td>
</tr>
<tr>
<td>$G_{17}$</td>
<td>1-10</td>
<td>RSI buy threshold</td>
<td>22.5</td>
</tr>
<tr>
<td>$G_{18}$</td>
<td>1-10</td>
<td>RSI sell threshold</td>
<td>72</td>
</tr>
<tr>
<td>$G_{19}$</td>
<td>1-10</td>
<td>ROC level</td>
<td>5.5</td>
</tr>
<tr>
<td>$G_{20}$</td>
<td>1-10</td>
<td>ROC short-term value</td>
<td>11</td>
</tr>
<tr>
<td>$G_{21}$</td>
<td>1-10</td>
<td>ROC long-term value</td>
<td>22</td>
</tr>
<tr>
<td>$G_{22}$</td>
<td>1-10</td>
<td>SO K variable value</td>
<td>8.25</td>
</tr>
<tr>
<td>$G_{23}$</td>
<td>1-5</td>
<td>SO D variable value</td>
<td>3</td>
</tr>
<tr>
<td>$G_{24}$</td>
<td>1-10</td>
<td>SO buy threshold</td>
<td>19.25</td>
</tr>
<tr>
<td>$G_{25}$</td>
<td>1-10</td>
<td>SO sell threshold</td>
<td>72</td>
</tr>
<tr>
<td>$G_{26}$</td>
<td>1-10</td>
<td>MACD short-term value</td>
<td>11</td>
</tr>
<tr>
<td>$G_{27}$</td>
<td>1-10</td>
<td>MACD long-term value</td>
<td>24.75</td>
</tr>
<tr>
<td>$G_{28}$</td>
<td>1-10</td>
<td>MACD signal line</td>
<td>8.25</td>
</tr>
<tr>
<td>$G_{29}$</td>
<td>1-10</td>
<td>BB time period value</td>
<td>19.25</td>
</tr>
<tr>
<td>$G_{30}$</td>
<td>1-5</td>
<td>BB deviations number</td>
<td>3</td>
</tr>
</tbody>
</table>
3.6.5 Sharpe Ratio

The Sharpe Ratio, introduced by William Sharpe in 1966 [113, 114] and later revised in 1994 [115], is an indicator that provides information on how far the yield of an investment lay over the risk-free interest rate, and at which volatility this yield was achieved. The Sharpe Ratio allows ex post comparison between investments. In effect, it measures the expected return above that of the risk-free interest rate per unit of risk, with volatility representing the risk measure. It is a popular measure in finance, however as Sherstov and Stone pointed out, it has seen little use in automated trading literature [117], particularly in comparison to other measures. The following description of the ex post Sharpe Ratio is based on the formulae in [115]. First, the difference between the historical return on the security over a given time period \( R_{Ft} \) to the return that would have been earned when invested in the risk free alternative \( R_{Bt} \), in this implementation this is equal to a fixed interest rate, is determined and referred to as \( D_t \) as in Equation 3.24. This establishes a measure of whether investing in a security, and therefore increasing exposure to risk, would have yielded better returns than investing in risk-free alternative investments.

\[
D_t \equiv R_{Ft} - R_{Bt} \tag{3.24}
\]

The average value of this excess return over the risk-free alternative is determined as in Equation 3.25 by taking the average of \( D_t \) over time period \( T \), and is referred to as \( \bar{D} \).
\[ \bar{D} \equiv \frac{1}{T} \sum_{t=1}^{T} D_t \]  

(3.25)

Finally, the ex post Sharpe Ratio is then calculated by dividing \( \bar{D} \) by the standard deviation of \( D_t \), referred to as \( \sigma_D \) and calculated as in Equations 3.26 and 3.27.

\[ \sigma_D \equiv \sqrt{\frac{\sum_{t=1}^{T} (D_t - \bar{D})^2}{T - 1}} \]  

(3.26)

\[ S_h \equiv \frac{\bar{D}}{\sigma_D} \]  

(3.27)

In terms of the agents’ genomes, significantly fewer genes are required for the Sharpe Ratio, comparable in number to those in the Capital Asset Pricing Model. \( G_1 \) represents the period over which returns are measured and is multiplied by the constant value of 7, making the time frame more representative of the periods considered in wider literature, and with \( G_2 \) representing the period over which the performance of returns is measured and in turn being multiplied by a constant value of 2 for the same reason. \( G_3 \) to \( G_7 \) perform the same function as in Section 3.6.3 for the Capital Asset Pricing Model. A breakdown of all the genes is shown in Table 3.6.

3.6.6 Random

Perhaps forming the most fundamental benchmark available, a random selection and fund allocation strategy can be used to determine if any of the methodologies described above offer benefits to trading agents. Simplistically, a random selection of
Table 3.6: Sharpe Ratio gene descriptions

<table>
<thead>
<tr>
<th>Gene</th>
<th>Value range</th>
<th>Description / Function</th>
<th>Local approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>1-10</td>
<td>Time frame</td>
<td>38.5</td>
</tr>
<tr>
<td>$G_2$</td>
<td>1-10</td>
<td>Time Period</td>
<td>11</td>
</tr>
<tr>
<td>$G_3$</td>
<td>1-10</td>
<td>Capital investment proportion</td>
<td>55</td>
</tr>
<tr>
<td>$G_4$</td>
<td>0-9</td>
<td>Minimum sell difference</td>
<td>4.5</td>
</tr>
<tr>
<td>$G_5$</td>
<td>0-9</td>
<td>Minimum buy difference</td>
<td>4.5</td>
</tr>
<tr>
<td>$G_6$</td>
<td>1-10</td>
<td>Securities to invest in (1)</td>
<td>5.5</td>
</tr>
<tr>
<td>$G_7$</td>
<td>1-10</td>
<td>Securities to invest in (2)</td>
<td>5.5</td>
</tr>
</tbody>
</table>

securities are bought and sold for a random amount on each trading day. No evolution or training of any sort takes place with performance measured following the same standards as the other methodologies.

3.7 Statistical analysis

Two statistical methods of analysing results were used, specifically the Mann-Whitney U test and the Kruskal-Wallis test. Both tests are non-parametric tests and compare the distribution of samples, where the Mann-Whitney U test always compares only two samples, and the Kruskal-Wallis test compares three or more samples. This section will introduce each test, with a more thorough explanation and mathematical illustrations provided in [118].

Throughout, significance in statistical analysis was taken at $p < 0.01$ and at a 99% confidence interval with all quoted p values being two sided, unless stated otherwise.
3.7.1 Mann-Whitney U test

The Mann-Whitney U test compares two samples, with the null hypothesis stating that both samples are drawn from the same population. In other words, no difference exists between the samples. In relation to this work, any experiment comparing results using the Mann-Whitney U test assumes that the samples are from the same population, with the alternative hypothesis stating that they are not.

The statistic calculates a value, referred to as $U$, which is derived by ranking results from both samples indiscriminately from which sample set they came from. In case of ties, an average value is assigned. The $U$ value is expressed mathematically:

$$U_a = n_a n_b + \frac{n_b(n_a + 1)}{2} - T_a$$  \hspace{1cm} (3.28)

where $U_a$ is the U value of sample $a$, $n$ the sample size for $a$ and $b$, and $T_a$ being the sum of assigned ranks in sample $a$.

3.7.2 Kruskal-Wallis test

As an extension of the Mann-Whitney U test, the Kruskal-Wallis test follows a similar approach. The null hypothesis stating that all samples are drawn from the same population and no difference exists between them. In relation to this work, any experiment comparing results using the Kruskal-Wallis test therefore uses a null hypothesis whereby results for each sample are similar, with the alternate stating that a difference exists between one or more of the samples.
The Kruskal-Wallis test ranks results in the same manner as the Mann-Whitney U test, however a key difference lies in that not only their sum and rankings are considered, but also the mean for each sample. The statistic allows for the calculation of a value, commonly referred to as $H$, which essentially represents the variances of ranks among samples, and is expressed mathematically as:

$$H = \frac{12}{N(N+1)} \sum_{h=1}^{g} \frac{S_h}{n_h} - 3(N+1)$$

(3.29)

where $N$ is the total number of observations across samples, $g$ the number of samples used, $S_h$ the square of sample $h$’s sum of ranks and $n_h$ the number of observations in sample $h$.

### 3.8 Summary

This section presented the final system implementation used as a basis for the experiments conducted as part of this research. The agent’s functionality representing traders with their own portfolio, capital and genome determining their analysis and trading behavior as well as their decision process were introduced. The permutations of Holland’s genetic algorithm and the various fitness types partially used throughout all experiments were presented in detail, with particular emphasis on FT1 representing the area under an agent’s total asset graph as well as FT3 representing the more common measure of profit. Lastly, the different methodologies used throughout and studied in depth in Chapter 4 were presented.
CHAPTER 4

An Investigation into the Use of Different Trading Methodologies

4.1 Introduction

This chapter investigates genetic algorithm evolved agents trading on real historical equity market data using technical analysis, the capital asset pricing model and a hybrid model of the two approaches. Three agent groups are generated, each using solely one of the two approaches or their hybrid to determine trading decisions. Each group consists of ten independently evolved populations over a thousand generations, whose elite’s performances are consequently averaged and used to compare to the other approaches.
4.2 System Outline

Agent populations emulating equity market traders were evolved using the oGA, with three methods of analysing securities implemented. Namely Technical Analysis, the Capital Asset Pricing Model and a hybrid approach of the previous two methods. As described earlier, each agent represents an individual trader with a personal portfolio and capital holdings, collectively referred to as their total assets. With a fixed starting capital, at the end of every trading day each trader uses historical price data to make a decision on every security whether to buy, sell or hold. No limitations, apart from capital constraints, are set on the number of trades conducted. Additionally, every agent contains a genome which specifies the parameters used by that particular agent for its analysis and decision processes, effectively determining its trading behavior. Depending on the method of analysis used, genome length varies with the number of parameters employed.

4.3 Method and Results

Data used in the simulation was presented in Section 3.3. Data set K, with 1465 trading days worth of data, was used representing six years from 01.01.1990 to 29.12.1995. Based on results presented later in Chapter 6, this time period was assumed sufficient to evolve a competitive population. The out-of-sample period, data set J, was taken over six years from 01.01.1997 to 31.12.2002. This time period further seemed
particularly suitable displaying both upward and downward trending characteristics, avoiding strategies biased towards either environment appearing incorrectly superior or inferior in analysis. As Doeksen et al pointed out, studies in classification problems have shown the importance of equal representations of both cases in order to prevent a bias towards the more common value, when commenting on day-to-day changes in closing price information [31].

For each approach, 10 populations of a 100 agents each were independently evolved from initially random populations over 1000 generations. The fitness function used the total area under the agent’s total asset graph over the entire training duration. The elite from each population was exposed to the testing data with their performance averaged and a mean taken across the 10 populations for each approach. Beta correlation to the buy-and-hold strategy was also measured, being defined as how closely the performance followed that of the buy-and-hold strategy, with a value of 1 indicating synchronous movement, 0 indicating no correlation and -1 indicating asynchronous movement. The performance’s volatility was also noted, with volatility being defined as the average daily relative change in value, giving an indication as to the fluctuations experienced on a day to day basis. Experiments were repeated for the two interest rates of 0% and 5% and are presented in Tables 4.1, 4.2, 4.3, with results summarized for convenience in Table 4.4.
### Table 4.1: Capital Asset Pricing Model Results

<table>
<thead>
<tr>
<th></th>
<th>Interest 0%</th>
<th></th>
<th></th>
<th>Interest 5%</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Performance (x10^8)</td>
<td>Beta</td>
<td>Volatility (%)</td>
<td>Performance (x10^8)</td>
<td>Beta</td>
<td>Volatility (%)</td>
</tr>
<tr>
<td>Buy-Hold</td>
<td>2.701</td>
<td>-</td>
<td>1.72</td>
<td>2.701</td>
<td>-</td>
<td>1.72</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 1</td>
<td>2.468</td>
<td>0.57</td>
<td>1.21</td>
<td>2.267</td>
<td>0.89</td>
<td>1.09</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 2</td>
<td>2.423</td>
<td>0.54</td>
<td>1.24</td>
<td>2.287</td>
<td>0.84</td>
<td>1.08</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 3</td>
<td>2.433</td>
<td>0.55</td>
<td>1.26</td>
<td>2.283</td>
<td>0.87</td>
<td>1.11</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 4</td>
<td>2.417</td>
<td>0.58</td>
<td>1.21</td>
<td>2.279</td>
<td>0.86</td>
<td>1.10</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 5</td>
<td>2.451</td>
<td>0.60</td>
<td>1.23</td>
<td>2.264</td>
<td>0.85</td>
<td>1.12</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 6</td>
<td>2.434</td>
<td>0.54</td>
<td>1.28</td>
<td>2.262</td>
<td>0.88</td>
<td>1.11</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 7</td>
<td>2.433</td>
<td>0.46</td>
<td>1.29</td>
<td>2.290</td>
<td>0.86</td>
<td>1.14</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 8</td>
<td>2.415</td>
<td>0.44</td>
<td>1.29</td>
<td>2.243</td>
<td>0.88</td>
<td>1.10</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 9</td>
<td>2.442</td>
<td>0.57</td>
<td>1.22</td>
<td>2.284</td>
<td>0.85</td>
<td>1.12</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 10</td>
<td>2.400</td>
<td>0.38</td>
<td>1.28</td>
<td>2.271</td>
<td>0.86</td>
<td>1.07</td>
</tr>
<tr>
<td>Average</td>
<td>2.431</td>
<td>0.52</td>
<td>1.25</td>
<td>2.273</td>
<td>0.86</td>
<td>1.10</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.190</td>
<td>0.07</td>
<td>0.03</td>
<td>0.150</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>Interest 0%</td>
<td></td>
<td></td>
<td>Interest 5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>-------------</td>
<td>----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>Performance (x10^8)</td>
<td>Beta</td>
<td>Volatility (%)</td>
<td>Performance (x10^8)</td>
<td>Beta</td>
<td>Volatility (%)</td>
</tr>
<tr>
<td>Buy-Hold</td>
<td>2.701</td>
<td>-</td>
<td>1.72</td>
<td>2.701</td>
<td>-</td>
<td>1.72</td>
</tr>
<tr>
<td>Technical Analysis 1</td>
<td>2.142</td>
<td>0.89</td>
<td>1.04</td>
<td>2.811</td>
<td>0.78</td>
<td>1.22</td>
</tr>
<tr>
<td>Technical Analysis 2</td>
<td>2.380</td>
<td>0.93</td>
<td>1.19</td>
<td>2.279</td>
<td>0.83</td>
<td>1.13</td>
</tr>
<tr>
<td>Technical Analysis 3</td>
<td>2.575</td>
<td>0.93</td>
<td>1.23</td>
<td>3.294</td>
<td>0.80</td>
<td>1.47</td>
</tr>
<tr>
<td>Technical Analysis 4</td>
<td>2.267</td>
<td>0.93</td>
<td>1.09</td>
<td>3.156</td>
<td>0.77</td>
<td>1.28</td>
</tr>
<tr>
<td>Technical Analysis 5</td>
<td>2.339</td>
<td>0.88</td>
<td>1.09</td>
<td>2.221</td>
<td>0.86</td>
<td>0.97</td>
</tr>
<tr>
<td>Technical Analysis 6</td>
<td>2.773</td>
<td>0.90</td>
<td>1.33</td>
<td>2.462</td>
<td>0.90</td>
<td>1.19</td>
</tr>
<tr>
<td>Technical Analysis 7</td>
<td>2.574</td>
<td>0.91</td>
<td>1.31</td>
<td>2.610</td>
<td>0.86</td>
<td>1.26</td>
</tr>
<tr>
<td>Technical Analysis 8</td>
<td>2.742</td>
<td>0.93</td>
<td>1.25</td>
<td>2.865</td>
<td>0.83</td>
<td>1.22</td>
</tr>
<tr>
<td>Technical Analysis 9</td>
<td>2.220</td>
<td>0.89</td>
<td>0.97</td>
<td>2.274</td>
<td>0.83</td>
<td>1.09</td>
</tr>
<tr>
<td>Technical Analysis 10</td>
<td>2.147</td>
<td>0.80</td>
<td>1.06</td>
<td>3.045</td>
<td>0.90</td>
<td>1.24</td>
</tr>
<tr>
<td>Average</td>
<td>2.416</td>
<td>0.90</td>
<td>1.16</td>
<td>2.702</td>
<td>0.83</td>
<td>1.21</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.236</td>
<td>0.04</td>
<td>0.12</td>
<td>0.391</td>
<td>0.04</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Table 4.3: Capital Asset Pricing Model and Technical Analysis hybrid results

<table>
<thead>
<tr>
<th></th>
<th>Performance (x10^8)</th>
<th>Beta</th>
<th>Volatility (%)</th>
<th>Performance (x10^8)</th>
<th>Beta</th>
<th>Volatility (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interest 0%</td>
<td>Buy-Hold</td>
<td>2.701</td>
<td>1.72</td>
<td>2.701</td>
<td>-</td>
<td>1.72</td>
</tr>
<tr>
<td>Interest 5%</td>
<td>Hybrid 1</td>
<td>2.493</td>
<td>0.97</td>
<td>2.822</td>
<td>0.72</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>Hybrid 2</td>
<td>2.495</td>
<td>0.88</td>
<td>2.744</td>
<td>0.83</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td>Hybrid 3</td>
<td>2.704</td>
<td>0.83</td>
<td>2.548</td>
<td>0.84</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>Hybrid 4</td>
<td>4.944</td>
<td>0.67</td>
<td>2.798</td>
<td>0.89</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td>Hybrid 5</td>
<td>2.436</td>
<td>0.81</td>
<td>3.598</td>
<td>0.62</td>
<td>1.48</td>
</tr>
<tr>
<td></td>
<td>Hybrid 6</td>
<td>3.053</td>
<td>0.84</td>
<td>2.705</td>
<td>0.81</td>
<td>1.04</td>
</tr>
<tr>
<td></td>
<td>Hybrid 7</td>
<td>2.575</td>
<td>0.78</td>
<td>2.582</td>
<td>0.91</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>Hybrid 8</td>
<td>2.485</td>
<td>0.92</td>
<td>2.891</td>
<td>0.90</td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td>Hybrid 9</td>
<td>2.905</td>
<td>0.87</td>
<td>2.715</td>
<td>0.86</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td>Hybrid 10</td>
<td>2.774</td>
<td>0.87</td>
<td>2.569</td>
<td>0.63</td>
<td>1.12</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>2.886</td>
<td>0.84</td>
<td>2.797</td>
<td>0.80</td>
<td>1.08</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td></td>
<td>0.751</td>
<td>0.08</td>
<td>0.304</td>
<td>0.11</td>
<td>0.17</td>
</tr>
</tbody>
</table>
## Table 4.4: Results Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Performance Mean (S.D. $\times 10^8$)</th>
<th>Beta Mean (S.D.)</th>
<th>Volatility Mean (S.D.) (%)</th>
<th>Sharpe Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy-Hold 0%</td>
<td>2.701</td>
<td>-</td>
<td>1.72</td>
<td>2.10</td>
</tr>
<tr>
<td>Buy-Hold 5%</td>
<td>2.701</td>
<td>-</td>
<td>1.72</td>
<td>1.75</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 0%</td>
<td>2.431 (0.019)</td>
<td>0.52 (0.07)</td>
<td>1.25 (0.03)</td>
<td>2.00</td>
</tr>
<tr>
<td>Capital Asset Pricing Model 5%</td>
<td>2.273 (0.015)</td>
<td>0.86 (0.02)</td>
<td>1.10 (0.02)</td>
<td>1.63</td>
</tr>
<tr>
<td>Technical Analysis 0%</td>
<td>2.416 (0.236)</td>
<td>0.90 (0.04)</td>
<td>1.16 (0.12)</td>
<td>2.78</td>
</tr>
<tr>
<td>Technical Analysis 5%</td>
<td>2.702 (0.391)</td>
<td>0.83 (0.04)</td>
<td>1.21 (0.13)</td>
<td>2.62</td>
</tr>
<tr>
<td>Hybrid 0%</td>
<td>2.886 (0.751)</td>
<td>0.84 (0.08)</td>
<td>1.17 (0.20)</td>
<td>2.48</td>
</tr>
<tr>
<td>Hybrid 5%</td>
<td>2.797 (0.304)</td>
<td>0.80 (0.11)</td>
<td>1.08 (0.17)</td>
<td>3.04</td>
</tr>
</tbody>
</table>
Further, the Capital Asset Pricing Model based approach was compared to the Technical Analysis based approach, the Capital Asset Pricing Model based approach to the hybrid approach and the Technical Analysis based approach to the hybrid approach, to establish the significance of the difference in results. Again, each of these tests were repeated for both 0% and 5%. Results using Mann-Whitney U tests are shown in Table 4.5.

4.4 Analysis and Discussion

Considering the variations in interest rates first, a clear impact can be observed for the Capital Asset Pricing Model agents. At an interest rate of 0%, agents performed significantly better than those at a rate of 5%. As interest rates are not considered in the decision making process in Technical Analysis, it is not surprising that no significant difference in results can be observed, despite a seemingly better performance at a 5% rate of interest. Similarly, no difference can be found for the hybrid model, with
mean performance marginally lower at a 5% rate of interest than at 0%. Notably, 
disregarding an outlier in results for the hybrid method at 0%, having returned a 
performance of 4.944x10^8, would result in a performance mean of 2.658x10^8 with a 
standard deviation of 0.216x10^8.

Comparing the individual methods to one another in each respective interest rate 
category, under an interest rate of 5% the populations using Technical Analysis clearly 
outperformed those using the Capital Asset Pricing Model, while the hybrid approach 
seemingly outperformed both Capital Asset Pricing Model and Technical Analysis 
based approaches. Using 0% however, little difference in performance exists between 
Capital Asset Pricing Model and Technical Analysis based approaches, while the 
hybrid approach performed considerably better. In terms of volatility and beta factor, 
little difference exists between all approaches at both interest rates, with the exception 
of the Capital Asset Pricing Model at 0% displaying an extremely low beta value.

Looking at both Tables 4.4 and 4.5, it appears that the Capital Asset Pricing 
Model based approach performs comparably or worse than the Technical Analysis 
based approach, while the Technical Analysis based approach can perform comparably 
or worse than the hybrid. The Capital Asset Pricing Model based approach always 
displayed worse performance than the hybrid. It is also important to note that the 
standard deviation observed for all three approaches varies significantly, with the 
greatest deviation experienced by the hybrid, while the Technical Analysis based 
approach in turn had a significantly higher deviation compared to the Capital Asset
Pricing Model based approach. In other words, consistency in performance dropped from the Capital Asset Pricing Model to Technical Analysis to the hybrid approach. Overall it appears that in this implementation, agent populations that utilize the Capital Asset Pricing Model as their method for generating input into their decision models will generate consistent if comparatively below average returns. Particularly compared to the buy-and-hold strategy Capital Asset Pricing Model agents do not seem to perform very well. The agent populations that used Technical Analysis to feed into their decision models perform comparatively to the buy-and-hold strategy, as also supported by work presented later in Chapter 6. Lastly, agents using the hybrid method proved clearly superior to the others, albeit significantly less consistent in their returns. With respect to the buy-and-hold strategy the hybrid method further appears marginally better, though this cannot be substantiated here.

In conclusion, the hybrid method employed demonstrated better performance than its constituents, suggesting further hybridization of alternative methods. Future work implementing systems with increased complexity combining methodologically heterogeneous agent groups might also demonstrate similar benefits as demonstrated by the hybrid method in this simulation, as well as reduce the impact of inconsistency in performance. In this series of investigations however, when limiting the system to a single methodology, technical analysis will be used as the default based on its wider use and allowing for easier comparison of results with other work.
4.5 Summary and Conclusions

This section outlined the system settings used to compare the different methodologies studied in this work and the results of their comparison.

A static system was used with multiple agent populations trained independently over the specified data range prior to being exposed to the out-of-sample data. All system settings were kept constant with the only difference lying in some agent populations using the Capital Asset Pricing Model, some Technical Analysis and some a hybrid of the two as their methodology. As a benchmark all were compared to the buy-and-hold strategy to provide additional context to the performance observed.

Results showed that the choice of an agent’s methodology clearly impacted on their performance and indicated that the technical analysis based approach performed better than the capital asset pricing model based approach, while the hybrid approach in turn outperformed both. The results would suggest significant benefits in performance oriented implementation of hybrid or multi-method based approaches in agent-based systems. Agent’s using the Capital Asset Pricing Model overall showed the most lacking performance, with agent’s using Technical Analysis demonstrating an overall better outcome. The hybrid method of Technical Analysis coupled with the Capital Asset Pricing Model performed best overall in comparison, suggesting hybridization of methods should allow for further performance increases.
CHAPTER 5

An Investigation into the Impact of

Genetic Algorithms

5.1 Introduction

Portfolio optimization and trading competition simulations have employed a variety of heuristics to find optimized solutions [131, 98, 63, 33, 111, 134, 125, 4]. In this chapter the effect of a series of popular permutations of a standard genetic algorithm publicised by Holland and their effect on the performance measure of agents trading on historical equity market data will be studied. The variations on the standard genetic algorithm were based on combinations of tournament and roulette selection, as well as the use of ranked fitness and fitness scaling. As a further extension to the experimentation series, a variety of fitness measures were implemented to reflect alternative methods of assessing a trader’s performance.

In this work, in order to vary only one element in each experiment run, experi-
5. AN INVESTIGATION INTO THE IMPACT OF GENETIC ALGORITHMS

mentation was divided into two separate units. First, the genetic algorithm employed was varied and tested across all implemented methodologies described in Section 3.6, with performance and convergence of the populations analysed to determine the algorithms influence or impact. In the second set of experiments the fitness measure was varied using the oGA and technical analysis as the methodology, analysing the impact it had on the agent populations’ performance.

Experimental design and results for variations in the genetic algorithm are presented in the following section. This is followed in turn by the experimental setup and results for variations in the fitness levels prior to some concluding remarks.

5.2 Genetic Algorithms Investigated

The performance of agent populations was monitored to determine the impact of using the variety of genetic algorithm implementations as described in Section 3.4. Agents were designed to emulate market traders using technical analysis.

5.2.1 Experiment Design

A population of 100 agents was evolved using each of the genetic algorithm variations over the specified data set K (see Section 3.3). This was repeated 10 times for each population for every one of the four methodologies implemented. The top 25% from each evolved population were then exposed to the out-of-sample data set J (see Section 3.3), with their performance measured using FT1.
5.2.2 Results

Three measures were taken in each experiment to analyse the agent populations, with performance representing the fitness function and its optimization being the objective. The other measures considered agent performance in training across generations as well as the speed at which the genome pool of each population converged to a particular dominant solution. Lastly, the CPU time taken for each algorithm to run was measured and compared.

*Testing Performance*

The mean testing performance for each algorithm of each of the 10 populations was compared across methodologies, with results shown in Table 5.1.
### Table 5.1: Results comparing agent performance by genetic algorithm used

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tech. Analysis (x10^8)</th>
<th>Hybrid (x10^8)</th>
<th>CAPM (x10^8)</th>
<th>Sharpe (x10^8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>oGA</td>
<td>2.407</td>
<td>0.420</td>
<td>2.323</td>
<td>0.745</td>
</tr>
<tr>
<td>sGA</td>
<td>2.188</td>
<td>0.451</td>
<td>2.223</td>
<td>0.525</td>
</tr>
<tr>
<td>tGA</td>
<td>2.353</td>
<td>0.351</td>
<td>2.389</td>
<td>0.677</td>
</tr>
<tr>
<td>rsGA</td>
<td>2.309</td>
<td>0.540</td>
<td>2.402</td>
<td>0.348</td>
</tr>
<tr>
<td>rsGAfs</td>
<td>2.321</td>
<td>0.256</td>
<td>2.390</td>
<td>0.561</td>
</tr>
<tr>
<td>rtGAf</td>
<td>2.482</td>
<td>0.303</td>
<td>2.599</td>
<td>0.802</td>
</tr>
</tbody>
</table>
Overall, all algorithms produced populations displaying similar performance and standard deviation values with only a few exceptions across all values. When comparing the average performance of each algorithm however, it appears that the rtGAs and rsGAs performed worse with respective mean values of 2.395x10^8 and 2.404x10^8. More mid-ranged performance can be observed by the rsGA, sGA and oGA with respective mean values of 2.459x10^8, 2.500x10^8 and 2.517x10^8. The best performance overall was showcased by the tGA with a mean value of 2.596x10^8. Note should also be taken of the results observed in the oGA and Sharpe combination, which displayed a comparatively insignificant standard deviation of 584. This would suggest that the elite evolved by the oGA was dominated by a single successful solution in every experiment run, unlike the elite populations evolved by the other algorithms.

*Training Performance*

During training, the mean performance of each population was measured for each generation during the evolutionary process. A graphical representation of the mean population’s performance for each algorithm is shown in Figures 5.1, 5.2, 5.3 and 5.4, split based on the methodology employed.
5. AN INVESTIGATION INTO THE IMPACT OF GENETIC ALGORITHMS

Figure 5.1: Genetic algorithms performance by generation using technical indicators
5. An investigation into the impact of genetic algorithms

![Genetic algorithms performance by generation using the hybrid approach](image)

**Figure 5.2**: Genetic algorithms performance by generation using the hybrid approach
Figure 5.3: Genetic algorithms performance by generation using the CAPM
Figure 5.4: Genetic algorithms performance by generation using the Sharpe ratio
Overall, the oGA performed best at the end of training for every training set, while the sGA demonstrated the worst performance across all generations. The remaining genetic algorithm implementations performed comparably to each other, generally remaining between oGA and sGA.

Training Genome Convergence

In order to determine a level of convergence of the agent populations’ genomes, each agent’s genome at every generational step was compared to a mean genome. This mean genome was determined by taking the mean value of each gene for the entire population. A correlation analysis was then performed for every agent’s genome in reference to the mean genome with the population’s average correlation coefficient used to indicate the overall level of convergence on a scale of 1 to -1. In other words, a value of 1 would indicate complete convergence, 0 would indicate no relationship between any genomes while -1 would theoretically suggest an inverse correlation relationship. Results are presented graphically in Appendix B, with a summarized overview presented in Figures 5.5, 5.6, 5.7 and 5.8.
5. AN INVESTIGATION INTO THE IMPACT OF GENETIC ALGORITHMS

Figure 5.5: Technical Indicators Genome Correlation Overview
Figure 5.6: Hybrid Genome Correlation Overview
Figure 5.7: CAPM Genome Correlation Overview
Figure 5.8: Sharpe Genome Correlation Overview
As is clearly apparent, the rtGAs and oGA are generally the slowest to converge on a particular solution, while the remaining algorithms tend to converge fairly quickly. Interestingly, the rtGAs also exhibits a far greater variation in genomes than the rsGAs, despite both algorithms making use of fitness scaling. A further observation is the number of generations required for populations to converge on any particular solution lengthens as the agents’ genomes length increases. The hybrid overview showcases this in particular, with the oGA taking approximately 400 generations prior to reaching a plateau comparable to the majority of the other algorithms. In contrast the CAPM overview shows all populations converged by approximately 80 generations, likely due to the significantly smaller genome size.

*Algorithm Speed*

To determine run-times of each algorithm, which may very well affect the choice of an algorithm in resource constraint applications, each algorithm was run over 500 iterations and timed to a millisecond resolution. This process was repeated 10 times for each algorithm and averaged, with results shown in Table 5.2. The computer used for these measurement was an Intel Pentium 4 (2.40GHz), running Windows XP Pro (SP2), with every effort having been made at eliminating any other processes during run-time that may affect the observations made.
5.2.3 Analysis

A statistical comparison was performed on the results obtained from testing and training. The results of these are again split into testing performance, training performance and the genome convergence observed in training.

Testing Performance

The performance of each algorithm was compared to each other across methodologies using the non-parametric Mann-Whitney U Test. Results are shown in Table 5.3.

As is evident from Table 5.3, no significant difference exists at all between algorithms in terms of performance at a 99% confidence interval. This suggests that all algorithms are able to evolve equally successful trading populations and are working fine.

Further tests can be performed using a rank based analysis, where the performance of each algorithm is ranked from 1 down to 6 with decreasing performance. This eliminates absolute differences in the results and provides for a relative comparison between them, where the algorithm with the lowest score evolved the overall
Table 5.3: Statistical analysis of algorithms across methodologies

<table>
<thead>
<tr>
<th>Tests compared</th>
<th>U value</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>oGA sGA</td>
<td>9.0</td>
<td>0.88571</td>
</tr>
<tr>
<td>oGA tGA</td>
<td>10.0</td>
<td>0.68571</td>
</tr>
<tr>
<td>oGA rsGA</td>
<td>11.0</td>
<td>0.48571</td>
</tr>
<tr>
<td>oGA rsGAfs</td>
<td>12.0</td>
<td>0.34286</td>
</tr>
<tr>
<td>oGA rtGAfs</td>
<td>10.0</td>
<td>0.68571</td>
</tr>
<tr>
<td>sGA tGA</td>
<td>10.0</td>
<td>0.68571</td>
</tr>
<tr>
<td>sGA rsGA</td>
<td>8.0</td>
<td>1.00000</td>
</tr>
<tr>
<td>sGA rsGAfs</td>
<td>8.0</td>
<td>1.00000</td>
</tr>
<tr>
<td>sGA rtGAfs</td>
<td>10.0</td>
<td>0.68571</td>
</tr>
<tr>
<td>tGA rsGA</td>
<td>10.0</td>
<td>0.68571</td>
</tr>
<tr>
<td>tGA rsGAfs</td>
<td>11.0</td>
<td>0.48571</td>
</tr>
<tr>
<td>tGA rtGAfs</td>
<td>11.0</td>
<td>0.48571</td>
</tr>
<tr>
<td>rsGA rsGAfs</td>
<td>10.0</td>
<td>0.68571</td>
</tr>
<tr>
<td>rsGA rtGAfs</td>
<td>9.0</td>
<td>0.88571</td>
</tr>
<tr>
<td>rsGAfs rtGAfs</td>
<td>10.0</td>
<td>0.68571</td>
</tr>
</tbody>
</table>

best populations. The algorithm with the highest score evolved the overall worst performing populations. Rankings and their overall score and rank are shown in Table 5.4.

Table 5.4: Rank based analysis of algorithms across methodologies

<table>
<thead>
<tr>
<th>Tech. Analysis</th>
<th>oGA</th>
<th>sGA</th>
<th>tGA</th>
<th>rsGA</th>
<th>rsGAfs</th>
<th>rtGAfs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CAPM</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Sharpe</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Sum</td>
<td>13</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>18</td>
<td>13</td>
</tr>
<tr>
<td>Sum Rank</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

As can be seen from Table 5.4, the tGA performed better relative to the other algorithms most of the time when compared across methodologies. Both oGA and rtGAfs performed similarly next with sGA and and rsGA however performing worse.
Following the ranked based analysis rsGAs performed worst.

Considering rankings for each algorithm, results appear inconsistent. The rtGAs for example performs best relative to the other algorithms for technical analysis and the hybrid approach, however does not appear to do so well when used with the capital asset pricing model and sharpe. Similar notable inconsistencies can be observed in the sGA.

Overall, no significant difference in results can be found between any of the algorithms. Using a rank based analysis tGA, oGA and rtGAs appear only marginally preferable to sGA, rsGA and rsGAs. In conclusion therefore any of the algorithms implemented will produce equally capable populations when evolved over 1000 generations.

*Training Performance*

Performance measures were taken at generation 100, 250, 500 and 1000 during training to compare each algorithm’s speed at evolving successful solutions. This measure in particular is of interest for adaptive systems, such as in Chapter 6.3, where significant savings can be made in terms of resources if an algorithm that produces good solutions in fewer generations is used. In this work however no time-sensitive aspect was included.

Looking at the graphs in Figures 5.1, 5.2, 5.3 and 5.4, the following relative ranking to each other split by algorithm and methodology was obtained in Table 5.5. The
values in the table compare each algorithm to one another in terms of performance at the four generation points, specifically again generations 100, 250, 500 and 1000. The algorithm that has produced the highest performing solution at each generational point is ranked first, the next performing ranked second and so on with the algorithm having produced solutions with the lowest performance being ranked 6th. This is repeated for each methodology. The rank obtained at each generation point across methodologies is then summed and compared to the sum of all other algorithms at the same generation point. For instance, the oGA at generation point 100 is compared to the sGA at generation point 100. The sums for the algorithms are 10 and 24 respectively, clearly suggesting that the oGA produced better solutions than the sGA at this point in evolution. All algorithms are then compared using their sums, with the lowest sums being ranked first and so on.

Out of 16 measurements taken for each algorithm the oGA was ranked first 14 times. As is clearly evident, the oGA therefore produces the best performance most rapidly overall, while the sGA fares worst in both respects. Between these, the tGA formed the second most successful algorithm followed by the rsGAs and rtGAs at a comparable level, leaving the rsGA marginally better than the sGA.

These observations therefore show that in training the best solutions for that particular data set are found by the oGA, as well as finding better solutions faster than the other algorithms.
5. AN INVESTIGATION INTO THE IMPACT OF GENETIC ALGORITHMS

Table 5.5: Rank based analysis of algorithms in training across methodologies

<table>
<thead>
<tr>
<th></th>
<th>Tech. Analysis</th>
<th>Hybrid</th>
<th>CAPM</th>
<th>Sharpe</th>
<th>Sum</th>
<th>Sum Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>oGA</td>
<td>at 100</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>at 250</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>at 500</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>at 1000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>sGA</td>
<td>at 100</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>at 250</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>at 500</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>at 1000</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>tGA</td>
<td>at 100</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>at 250</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>at 500</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>at 1000</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>rsGA</td>
<td>at 100</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>at 250</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>at 500</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>at 1000</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>rsGAs</td>
<td>at 100</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>at 250</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>at 500</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>at 1000</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>rtGAs</td>
<td>at 100</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>at 250</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>at 500</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>at 1000</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>13</td>
</tr>
</tbody>
</table>

Training Genome Convergence

Considering the overall speed of convergence of each algorithm in Figures 5.5, 5.6, 5.7 and 5.8, clear differences can be observed.

The oGA tends to converge gradually but generally at a significantly slower pace than the majority of the other algorithms. In particular the rsGA and tGA tend to converge rapidly across all methodologies. The sGA appears half way between the
three, converging more rapidly than the oGA but not reaching a plateau as quickly as the rsGA or tGA. The algorithms using fitness scaling distinguish themselves by operating at a lower correlation coefficient than the sGA, tGA and rsGA, a result of their emphasis on encouraging diversity in the populations. The oGA varies between the two groups in that respect, where it converges to a coefficient of just below 1 in the hybrid, technical analysis and capital asset pricing model systems, but considerably lower in the Sharpe based system.

Overall the rtGAs in particular and to some extend the rsGAs are clearly most desirable when aiming to maintain a diverse population set. When considered looking at performance results, the rtGAs in particular distinguishes itself by evolving generally better results than the other algorithms. The tGA, rsGA and rsGAs demonstrated the fastest rate of conversion, finding competitive solutions with far fewer generational runs making them computationally inexpensive as an approach. Comparatively, agent populations in these algorithms converged within 10–20 generations compared to 100–400 in case of the oGA for example. What is important to consider however is that the solutions found more rapidly in the case of the rsGA are clearly inferior based on Table 5.5, while the tGA on the other hand found highly competitive solutions in the same time span. From this perspective the tGA seems to suggest the computationally least expensive algorithm to evolve competitive agents. Overall however, it could be argued that with a few additional generations performance considerably better solutions can be found by the oGA for instance (see Figures 5.1
Algorithm Speed

As can be clearly seen from Table 5.2, the oGA displayed the shortest run time for 500 iterations averaged across 10 independent measurements. The sGA and tGA displayed comparable times, with the remaining algorithms requiring slightly more time. This is likely a result of the increasing complexity of those algorithms. The difference between algorithms overall however, except for the oGA, were minimal. In comparison, all algorithms appeared significantly slower than the oGA. Following from the earlier observation of fewer generations being required to evolve a competitive solution using the tGA, in a resource constrained system it may be that the increased computational expense of using the tGA may negate the apparent advantage over the oGA. In conclusion, based on actual computational time required, the oGA implementation is the fastest algorithm observed in this investigation.

5.2.4 Conclusions

It was found that no or at most negligible differences exist in performance between solutions evolved by the six different algorithms when tested across the four methodologies described. In terms of training performance however the oGA clearly performed better than all other algorithms, as well as finding better solutions more rapidly. In terms of convergence, the tGA clearly outperformed all algorithms, converging to
successful solutions far more rapidly than the other algorithms. The computationally least expensive algorithm by a considerable margin, displaying the shortest time required to run over 500 iterations, was the oGA.

The oGA distinguishes itself from the other algorithms due to several of its characteristics, using elitism, immigration and no mutation. Which factor or combination of factors determined the better performance in training however cannot be pinpointed without further investigation.

Though in a static system with abundant resources any of the algorithms would appear equally suitable, it would appear that particularly in adaptive, time critical or resource constraint systems using the oGA or perhaps tGA may offer benefits over the other genetic algorithms implemented.

5.3 Fitness Types Investigated

The performance of agent populations emulating market traders using technical analysis was monitored to determine the impact of using a variety of fitness measures, as described in Section 3.5. The fitness measure used in any optimization problem determines the nature of the solutions obtained, and could therefore have a significant impact on the performance of the evolved agent populations in this investigation. Using a pre-determined measure of performance as defined in Section 3.5.1 to evaluate and compare all the fitness measures should provide insight into their effectiveness. The best performing fitness type, if any, could then also provide indication as to what
fundamentals may affect the performance of agents emulating traders.

5.3.1 Experiment Design

Five fitness types, including the comparative performance measure equivalent to FT1, were run using all six algorithms but using only technical analysis as the methodology. Each experiment was run over 1000 generations using 100 agents and repeated 10 times using data set K for training and data set J for out-of-sample testing again (see Section 3.3). Recorded performance on the out-of-sample testing data however was measured using FT1 in each case to allow for comparison.

5.3.2 Results

The mean performance from the 10 runs for each combination of settings was taken with the average and standard deviation shown in Table 5.6. Performance shows the average area under the elite agents’ performance curve from each of the 10 runs, while standard deviation is included to give an indication of how consistently the populations fared.
Table 5.6: Mean performance by fitness type and algorithm

<table>
<thead>
<tr>
<th></th>
<th>oGA</th>
<th>sGA</th>
<th>tGA</th>
<th>rsGA</th>
<th>rsGAs</th>
<th>rtGAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.175</td>
<td>2.207</td>
<td>1.951</td>
<td>2.184</td>
<td>2.100</td>
<td>2.039</td>
</tr>
<tr>
<td></td>
<td>0.456</td>
<td>0.494</td>
<td>0.346</td>
<td>0.615</td>
<td>0.472</td>
<td>0.474</td>
</tr>
<tr>
<td>FT2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.736</td>
<td>2.421</td>
<td>1.881</td>
<td>2.0254</td>
<td>1.954</td>
<td>1.842</td>
</tr>
<tr>
<td></td>
<td>0.653</td>
<td>0.500</td>
<td>0.329</td>
<td>0.420</td>
<td>0.430</td>
<td>0.247</td>
</tr>
<tr>
<td>FT3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.575</td>
<td>2.100</td>
<td>2.483</td>
<td>2.189</td>
<td>2.113</td>
<td>2.283</td>
</tr>
<tr>
<td></td>
<td>0.656</td>
<td>0.503</td>
<td>0.639</td>
<td>0.441</td>
<td>0.524</td>
<td>0.286</td>
</tr>
<tr>
<td>FT4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.380</td>
<td>2.084</td>
<td>2.242</td>
<td>2.340</td>
<td>2.711</td>
<td>2.362</td>
</tr>
<tr>
<td></td>
<td>0.368</td>
<td>0.425</td>
<td>0.366</td>
<td>0.497</td>
<td>0.779</td>
<td>0.441</td>
</tr>
<tr>
<td>FT5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.116</td>
<td>2.340</td>
<td>2.190</td>
<td>2.408</td>
<td>2.435</td>
<td>2.436</td>
</tr>
<tr>
<td></td>
<td>0.151</td>
<td>0.305</td>
<td>0.369</td>
<td>0.323</td>
<td>0.287</td>
<td>0.211</td>
</tr>
</tbody>
</table>
To allow easier comparison between each fitness type prior to further analysis, an average taken across all results is shown in Table 5.7.

<table>
<thead>
<tr>
<th></th>
<th>FT1 Performance</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT1</td>
<td>2.110</td>
<td>0.476</td>
</tr>
<tr>
<td>FT2</td>
<td>1.977</td>
<td>0.332</td>
</tr>
<tr>
<td>FT3</td>
<td>2.291</td>
<td>0.508</td>
</tr>
<tr>
<td>FT4</td>
<td>2.353</td>
<td>0.480</td>
</tr>
<tr>
<td>FT5</td>
<td>2.321</td>
<td>0.274</td>
</tr>
</tbody>
</table>

As Table 5.7 has made more succinct, FT3, FT4 and FT5 all performed similarly, with FT5 showing a greater consistency in results as indicated by the lower averaged standard deviations. FT1 and particularly FT2 on the other hand appear to be faring worst across the experiments with a comparable average standard deviation as the other fitness types.

5.3.3 Analysis

To improve understanding of the results obtained, two methods of analysis were performed to determine if any of the fitness measures appear to return better results than the others. First, a statistical analysis using the non-parametric Mann-Whitney U Test will be conducted followed by a ranked analysis of the performance results, as introduced earlier.
Each set of six results for each fitness types, as shown in Table 5.6, is compared to each other fitness type in turn. This will establish the probable degree to which a difference exists between each fitness type as well as provide an indication as to whether some fitness types can be classified as a grouping to generally perform better than another. Results of the Mann-Whitney U Test are shown in Table 5.8.

<table>
<thead>
<tr>
<th>Tests compared</th>
<th>U value</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT5 FT4</td>
<td>19</td>
<td>0.93723</td>
</tr>
<tr>
<td>FT5 FT3</td>
<td>21</td>
<td>0.69913</td>
</tr>
<tr>
<td>FT5 FT2</td>
<td>32</td>
<td>0.02597</td>
</tr>
<tr>
<td>FT5 FT1</td>
<td>32</td>
<td>0.02597</td>
</tr>
<tr>
<td>FT4 FT3</td>
<td>21</td>
<td>0.69913</td>
</tr>
<tr>
<td>FT4 FT2</td>
<td>31</td>
<td>0.04113</td>
</tr>
<tr>
<td>FT4 FT1</td>
<td>32</td>
<td>0.02597</td>
</tr>
<tr>
<td>FT3 FT2</td>
<td>32</td>
<td>0.02597</td>
</tr>
<tr>
<td>FT3 FT1</td>
<td>29</td>
<td>0.09307</td>
</tr>
<tr>
<td>FT2 FT1</td>
<td>28</td>
<td>0.13203</td>
</tr>
</tbody>
</table>

The majority of results indicate that no significant difference exists between the fitness types. However, results where FT5 or FT4 are compared to FT2 and FT1 and results where FT3 is compared to FT2 respectively in turn have a considerably lower p value compared to the other results, suggesting a more noticeable difference between the five pairs. More specifically, as both FT5 and FT4 show no significant difference between each other but both demonstrate a more notable difference to FT1 and FT2, this observation suggests that the populations using FT4 and FT5 produced better performance. As FT1 and FT2 also do not show any significant difference between themselves in turn, it could be argued that FT4 and FT5 produce similarly successful
solutions while FT1 and FT2 produce similarly less successful solutions. FT3 on the other hand showed no significant difference to any other fitness type except to FT2 at a 95% confidence interval, suggesting it’s performance level is likely to sit between the two groups of FT1/FT2 and FT4/FT5.

Relating these observations back to Table 5.7, the remarks are mirrored in each case with FT4 and FT5 producing apparently better solutions while FT1 and FT2 seem to be faring worst out of all the fitness types analysed.

Using rank based analysis, Table 5.6 can be converted to show the following, with the sum of each fitness type’s rank and their position relative to the others shown for convenience in Table 5.9

<table>
<thead>
<tr>
<th></th>
<th>FT1 Rank</th>
<th>FT2 Rank</th>
<th>FT3 Rank</th>
<th>FT4 Rank</th>
<th>FT5 Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>oGA</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>sGA</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>tGA</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>rsGA</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>rsGAfs</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>rtGAfs</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Sum 22</th>
<th>26</th>
<th>15</th>
<th>14</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum Rank</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

After ranking, FT3, FT4 and FT5 are all very close with each having one or two top results within an algorithm. FT1 and FT2 on the other hand fared considerably worse, with FT2 in particular standing out as the worst fitness type for algorithms except for an apparently anomalous result using the sGA. If standard deviations were
also ranked, FT5 would still remain the best followed by FT2, FT3, FT1 and as the least consistent FT4.

Placed in context with the previous analyses, the observations are well in line with their findings. FT4 and FT5 are ranked best with FT1 and FT2 ranked lowest. FT3 in this case appears much closer to FT4 and FT5 than was apparent by the statistical analysis, however the close proximity was also reflected in Table 5.7.

5.3.4 Conclusions

Five measures of fitness were implemented and employed in simulations using the six different genetic algorithms introduced earlier, all using the technical indicators as the trading methodology. Performance was measured using FT1 as it appeared the most reflective general measure of performance, despite the possible bias this might have created.

Results overall appear similar, though in analysis more apparent differences can be observed. In terms of performance, using FT4 and FT5 will produce solutions that perform the best. As FT5 also performs well in terms of consistency, it is the most desirable choice to produce populations that will perform well and consistently across different genetic algorithms. In this series of investigations however, FT1 will be used as both fitness measure and for comparison as it is the arguably most reflective general measure of performance.
5.4 Summary and Conclusions

Five measures of fitness and six different genetic algorithms were tested, all using technical indicators for their trading methodology. FT1 was used as the most general measure of performance.

Using FT4 and FT5 will produce the best performing solutions. FT5 also demonstrated the highest level of consistent returns, suggesting the best choice to produce populations that will perform well and consistently across different genetic algorithms.

No or at most negligible differences exist in performance between the solutions evolved and tested across the methodologies. For training performance the oGA performed best in addition to finding better solutions more rapidly. The tGA however clearly outperformed all algorithms converging to successful solutions far more rapidly than the other algorithms.

Particularly in adaptive, time critical or resource constraint systems using the oGA or perhaps tGA would therefore offer significant benefits over the other algorithms explored.

Overall therefore results here support the sheer abundance of different algorithms and fitness measures used in computational finance. Though covering a large scale in terms of performance and other attributes, most research has had similar findings despite their at times fundamentally different approaches. Some of these will offer some benefits relative to one another in direct comparison, such as FT4 or FT5 and the oGA or tGA perhaps in this work, however most systems will offer a competitive
solution in their specific combination of methods, algorithms and overall approaches.

From this work, it appears that though some benefits may be gained from some specific implementations for a particular problem, such as designing the system specifically for a dynamic environment using an adaptive approach, and individual validation of each implementation will always be required as no general rules seem to be deduceable.
Chapter 6

An Investigation into Static and Adaptive Systems

6.1 Introduction

This chapter will present a comparison of static versus adaptive trading systems. As both systems are often used and well justified in wider literature, this section aims to establish a clearer insight whether either can provide any benefits over the other in a trading simulation such as this. To achieve this, a variety of experiments are performed to optimize both systems prior to performing a direct comparison where all factors are kept constant with the exception of one using the static approach and the other the adaptive approach. Results from each are then compared independently to a benchmark as well as relative to each other.
6.2 Static System

6.2.1 Fundamental Parameter Analysis

This section studies the impact of a number of different training settings on agent performance in the test environment. It is conjectured that due to the nature of financial time series, solutions can very easily suffer from over-fit. If knowledge of the final application environment is available in advance, a high level of similarity in training may on the other hand be desirable. The disadvantages of over-fit are particularly apparent when training and testing environments are discontinuous. In a continuous environment, such as where testing data follows training data closely, change generally occurs gradually, providing some reference context to the preceding circumstance suggesting a good solution in training will fair well in testing. Should more discontinuous environments however be considered, which may still occur even if testing data follows training data in close proximity, no such reference may exist and adversely affect the solutions performance in testing. Ultimately it would be preferable to obtain a more general solution able to operate effectively, if not optimally, in a multitude of different and changing environments. To investigate this, the following conjectures are formulated:

Conjecture 1 (C1): A solution evolved on limited training data will be highly specific to the small variety of situations encountered in that environment and fare considerably worse in an environment of larger variety.
Conjecture 2 (C2): A solution evolved using large amounts of training data will be fairly general in nature as it will have encountered a multitude of different environments, making it able to cope with changing environments.

Conjecture 3 (C3): Increasing the quantity of training data will improve the solution’s generality and robustness up to a point, after which no significant improvements can be made.

In other words, with respect to C1 and C2, a highly specialized solution is likely to fare worse when exposed to out-of-sample data than a more general solution that should perform moderately well across the board. With respect to C3, increasing the quantity of training data eventually leads to diminishing returns on performance and increasing it further will not improve the solution resulting in a form of plateau, as illustrated in Figure 6.1.

![Figure 6.1: Diminishing returns of training](image)
Experiment Design

For consistency, all training and testing was performed using identical data sets, defined as Training A–C and Testing D–F, as presented earlier in Section 3.3 and detailed in Table 6.1 for convenience.

### Table 6.1: Training and testing data

<table>
<thead>
<tr>
<th></th>
<th>Time period</th>
<th>Trading days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training A</td>
<td>01.01.88 to 29.12.89</td>
<td>521</td>
</tr>
<tr>
<td>Training B</td>
<td>01.01.85 to 29.12.89</td>
<td>1304</td>
</tr>
<tr>
<td>Training C</td>
<td>01.01.80 to 29.12.89</td>
<td>2609</td>
</tr>
<tr>
<td>Test D</td>
<td>01.01.90 to 31.12.91</td>
<td>522</td>
</tr>
<tr>
<td>Test E</td>
<td>01.01.90 to 30.12.94</td>
<td>1305</td>
</tr>
<tr>
<td>Test F</td>
<td>01.01.90 to 31.12.99</td>
<td>2610</td>
</tr>
</tbody>
</table>

With the three variant factors of population, generations/time and shares, only five experiments were run as part of this preliminary investigation, analysing the impact of each factor used in training on agent performance in testing. The experiments are outlined in Table 6.2, with the altered factor in bold.

### Table 6.2: Experiment description

<table>
<thead>
<tr>
<th></th>
<th>Population</th>
<th>Generations</th>
<th>Time</th>
<th>Shares</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1000</td>
<td>Variable</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>Variable</td>
<td>24 hrs</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>1000</td>
<td>Variable</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
<td>1000</td>
<td>Variable</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>1000</td>
<td>Variable</td>
<td>10</td>
</tr>
</tbody>
</table>

For every experiment three groups of agents were trained on the three training data sets A–C, resulting in 15 trained groups of agents. The elite from each group of agents was subsequently exposed to test data sets D, E and F. For each test, the
average performance of the elite in terms of profit at the end of the time period from training is recorded as a percentage. Additionally, the area under the daily total average assets graph is recorded as the more effective measure of performance. These measurements were done for every group in every test set, with the DAX-30 Index used as a comparative reference. Based on the conjectures outlined earlier, the following hypotheses can now be formulated:

Hypothesis 1 (H1): Populations evolved on data set A will fare worse than populations evolved on data sets B and C.

Hypothesis 2 (H2): Populations evolved on data sets B and particularly C will perform similarly across testing data sets demonstrating no particular affinity to any one set.

Hypothesis 3 (H3): Populations evolved on data set B should show little or no difference in performance in testing to populations evolved on data set C.

The hypotheses claim that solutions trained on little data are very specific to their environment and that solutions trained on large quantities of data are more general and only moderately successful across multiple environments. If this is true, then it should be possible to make the same observations in every experiment for every group of trained agents once exposed to test data.

Additionally, the elite’s performance in testing of agents evolved on data sets A, B and C were compared to the Index’s performance over the same time period as
a general benchmark, indicating whether they performed comparatively well to the market overall.

**Results**

Detailed experimental results are shown in Table 6.3. Comparing the values in each column to the DAX, as summarized by their mean, in the majority of cases the elite population outperformed the Index. The main exception relates to results from test F, where every agent group failed to attain comparative gains to the DAX and did not consistently manage to achieve better performance.

Table 6.3: Results

<table>
<thead>
<tr>
<th>Group 1-A</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 1-B</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 1-C</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-11.81</td>
<td>16.40</td>
<td>126.16</td>
<td>0.388</td>
<td>1.186</td>
<td>3.400</td>
<td></td>
<td></td>
<td>0.404</td>
<td>1.210</td>
<td>3.681</td>
</tr>
<tr>
<td>Group 1-B</td>
<td>-11.01</td>
<td>15.71</td>
<td>178.60</td>
<td>0.404</td>
<td>1.210</td>
<td>3.681</td>
<td></td>
<td></td>
<td>0.429</td>
<td>1.256</td>
<td>3.796</td>
</tr>
<tr>
<td>Group 1-C</td>
<td>-06.70</td>
<td>19.65</td>
<td>173.46</td>
<td>0.429</td>
<td>1.256</td>
<td>3.796</td>
<td></td>
<td></td>
<td>0.429</td>
<td>1.256</td>
<td>3.796</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 2-A</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 2-B</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 2-C</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-14.66</td>
<td>10.69</td>
<td>125.46</td>
<td>0.373</td>
<td>1.135</td>
<td>3.282</td>
<td></td>
<td></td>
<td>0.408</td>
<td>1.262</td>
<td>4.041</td>
</tr>
<tr>
<td>Group 2-B</td>
<td>-09.10</td>
<td>26.96</td>
<td>219.79</td>
<td>0.408</td>
<td>1.262</td>
<td>4.041</td>
<td></td>
<td></td>
<td>0.407</td>
<td>1.237</td>
<td>3.822</td>
</tr>
<tr>
<td>Group 2-C</td>
<td>-10.28</td>
<td>19.95</td>
<td>187.12</td>
<td>0.407</td>
<td>1.237</td>
<td>3.822</td>
<td></td>
<td></td>
<td>0.420</td>
<td>1.309</td>
<td>3.995</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 3-A</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 3-B</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 3-C</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-21.05</td>
<td>07.42</td>
<td>178.38</td>
<td>0.360</td>
<td>1.087</td>
<td>3.370</td>
<td></td>
<td></td>
<td>0.392</td>
<td>1.246</td>
<td>3.823</td>
</tr>
<tr>
<td>Group 3-B</td>
<td>-08.31</td>
<td>22.93</td>
<td>150.84</td>
<td>0.392</td>
<td>1.246</td>
<td>3.823</td>
<td></td>
<td></td>
<td>0.420</td>
<td>1.309</td>
<td>3.995</td>
</tr>
<tr>
<td>Group 3-C</td>
<td>-01.92</td>
<td>29.88</td>
<td>188.65</td>
<td>0.420</td>
<td>1.309</td>
<td>3.995</td>
<td></td>
<td></td>
<td>0.420</td>
<td>1.309</td>
<td>3.995</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 4-A</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 4-B</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 4-C</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-18.99</td>
<td>00.78</td>
<td>145.30</td>
<td>0.414</td>
<td>1.283</td>
<td>4.288</td>
<td></td>
<td></td>
<td>0.458</td>
<td>1.509</td>
<td>5.266</td>
</tr>
<tr>
<td>Group 4-B</td>
<td>-11.81</td>
<td>21.96</td>
<td>172.52</td>
<td>0.458</td>
<td>1.509</td>
<td>5.266</td>
<td></td>
<td></td>
<td>0.416</td>
<td>1.230</td>
<td>3.743</td>
</tr>
<tr>
<td>Group 4-C</td>
<td>-09.02</td>
<td>19.01</td>
<td>183.96</td>
<td>0.416</td>
<td>1.230</td>
<td>3.743</td>
<td></td>
<td></td>
<td>0.416</td>
<td>1.230</td>
<td>3.743</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 5-A</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 5-B</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>Group 5-C</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-13.99</td>
<td>17.75</td>
<td>177.87</td>
<td>0.435</td>
<td>1.479</td>
<td>4.879</td>
<td></td>
<td></td>
<td>0.450</td>
<td>1.536</td>
<td>5.072</td>
</tr>
<tr>
<td>Group 5-B</td>
<td>-08.56</td>
<td>30.06</td>
<td>166.82</td>
<td>0.450</td>
<td>1.536</td>
<td>5.072</td>
<td></td>
<td></td>
<td>0.438</td>
<td>1.520</td>
<td>5.492</td>
</tr>
<tr>
<td>Group 5-C</td>
<td>-08.86</td>
<td>38.39</td>
<td>225.43</td>
<td>0.438</td>
<td>1.520</td>
<td>5.492</td>
<td></td>
<td></td>
<td>0.438</td>
<td>1.520</td>
<td>5.492</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mean</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
<th>DAX</th>
<th>Test D</th>
<th>Test E</th>
<th>Test F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.396</td>
<td>1.213</td>
<td>3.700</td>
<td></td>
<td>0.376</td>
<td>1.187</td>
<td>3.925</td>
</tr>
</tbody>
</table>
Analysis and Discussion

Several Mann-Whitney U tests were used to statistically assess the hypotheses using agent performance for comparison. For each Test D, Test E and Test F, a comparison was made between groups evolved on data set A to groups evolved on data set C (A to C), groups evolved on data set A to groups evolved on data set B (A to B) as well as groups evolved on data set B to groups evolved on data set C (B to C). In total 9 tests were conducted, with the null hypothesis stating that no difference exists between each group’s performance, while the alternative hypothesis stating that the latter class of group performed better. For example, when comparing groups from training A to groups from training C, if the null hypothesis is rejected, performance from groups evolved on data set C will be considered significantly higher than that of groups evolved on data set A.

<table>
<thead>
<tr>
<th>Comparing:</th>
<th>A to C</th>
<th>A to B</th>
<th>B to C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test D</td>
<td>0.00397</td>
<td>0.00397</td>
<td>0.07540</td>
</tr>
<tr>
<td>Test E</td>
<td>0.00397</td>
<td>0.00397</td>
<td>0.21032</td>
</tr>
<tr>
<td>Test F</td>
<td>0.00397</td>
<td>0.00397</td>
<td>0.21032</td>
</tr>
</tbody>
</table>

As seen in Table 6.4, groups from training B and C are always significantly different from those of training A, while there exists no significant difference between groups from B or C in any of the tests.

Considering results from test D, a significant number of groups outperformed the Index in both measures of performance. Notably, not a single time does a group from
training A outperform a group from trainings B or C. Furthermore, all groups from training A, apart one, are the only ones that fail to outperform the Index, indicating that H1 is true.

In test E, as for test D, groups trained from training A again fail to be very competitive. Additionally, there is no apparent superiority between groups from training B to groups from training C.

For test F the most significant difference to previous tests is the failure of any agent group to outperform the Index, and only when considering performance in terms of area under the curve did groups 2-B, 3-C and 5-C post better results. However, groups from training C appear to perform better in the latter three experiments than those from training A or B. This is in support of H2, as it hypothesized that the most general solutions, in this case groups from training C, should fare best in the most varied environment, in this case test C, as it will be least specific to its training environment.

These results suggest that extensive training periods appear more desirable than short-term training, indicating a high level of robustness and performance, as suggested in H2. However, as performance of groups from training B and C are not significantly different it can be inferred that H3 is also true, as doubling data quantity from B to C had no significant effect.

H1 and H2 are confirmed, as groups of agents with fairly general solutions managed to outperform groups with more specific solutions in tests where the trading
environment varied significantly, regardless of any training related settings found in the five different experiments (see Table 6.2). H3 is also confirmed, as increasing training data is subject to diminishing returns on agents performance as no difference was exhibited by groups from training B or C, and a significant amount of training time could be saved if this is taken into account for future experiments.

Lastly, the impact of varying population size, generations run and the number of securities used can be considered as shown in Table 6.2. Comparing experiment series 1 to 2 first, there is no apparent difference in results. Using a ranked comparison among tests and training groups, series 1 to 2 showed no significant difference results. Similarly, comparing experiments series 1, 3 and 4, little difference is apparent. Again using a ranked comparison among test and training groups, series 1 and 3 show no significant difference while series 4 repeatedly performed worse than the other two, suggesting that the larger population size used had a negative effect on performance. Lastly, comparing experiment series 1 to series 5, series 5 appears to demonstrate better performance than series 1. Using the same ranked comparison again, no significant difference was found between series 1 and 5. In conclusion, the alternative population sizes used or evolutionary cycles run in this case affected performance only for the largest population size tested. Reducing the number of securities resulted in no difference in performance of the agent population. Similarly, changing the number of generations to be time bound rather than fixed did not affect performance either.
6.2.2 Evolved Agents versus Fixed Agents

To determine whether an evolved agent population offers any benefits, its performance was compared to that of a population of agents with fixed technical analysis parameters as recommended by literature. Specifically, nine agents, that were not trained but designed with a fixed genome, were tested in an identical fashion to the evolved population on the same set of testing data as in Section 6.2.1. This was done to establish an internal comparative measure to determine whether optimization of the genome and possible deviation from parameters recommended by literature for technical analysis lead to any improvements in performance. It was also assumed that the gene $G_3$, responsible for allocation of funds among purchases, is set to 7, representing an investment of 70% each time. Lastly, weightings were assumed to be equal for every indicator. A brief description of each agent and its results are shown in Table 6.5. Of particular note is Fixed 1, which did not invest at all in any securities retaining its full initial capital allocation over the entire testing duration. The remaining fixed agents represent variations of the decision types adoptable by the evolved agents.
Table 6.5: Results for fixed agents

<table>
<thead>
<tr>
<th>Description</th>
<th>Profit (%)</th>
<th>Performance (x10^6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test D</td>
<td>Test E</td>
</tr>
<tr>
<td>Fixed 1 'No-trade' strategy</td>
<td>00.00</td>
<td>00.00</td>
</tr>
<tr>
<td>Fixed 2 DT1 and risk-averse</td>
<td>-20.24</td>
<td>-08.13</td>
</tr>
<tr>
<td>Fixed 3 DT2 and risk-averse</td>
<td>-18.43</td>
<td>04.84</td>
</tr>
<tr>
<td>Fixed 4 DT3 and risk-averse</td>
<td>-19.71</td>
<td>-09.98</td>
</tr>
<tr>
<td>Fixed 5 DT4 and risk-averse</td>
<td>-21.04</td>
<td>-16.62</td>
</tr>
<tr>
<td>Fixed 6 DT1 and risk taking</td>
<td>-01.88</td>
<td>25.24</td>
</tr>
<tr>
<td>Fixed 7 DT2 and risk taking</td>
<td>-19.97</td>
<td>04.59</td>
</tr>
<tr>
<td>Fixed 8 DT3 and risk taking</td>
<td>-10.28</td>
<td>09.00</td>
</tr>
<tr>
<td>Fixed 9 DT4 and risk taking</td>
<td>-05.13</td>
<td>07.28</td>
</tr>
</tbody>
</table>
When compared to Table 6.3, it appears that the majority of results are worse than those of the evolved agents. A statistical comparison was performed using Mann-Whitney U tests to statistically determine the significance of those differences, with results shown in Table 6.6.

**Table 6.6: Statistical analysis of evolved versus fixed agents**

<table>
<thead>
<tr>
<th></th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test D</td>
<td>0.7429</td>
</tr>
<tr>
<td>Test E</td>
<td>0.1141</td>
</tr>
<tr>
<td>Test F</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

As is evident, no significant difference was found between the performance shown in Tables 6.3 and 6.5 for Tests D and E, while a significant difference existed for Test F. Though the evolved agents only appear more successful in one out of three tests, it is important to note that comparison included weak results from Training A, whose agents consistently fared worse than those from Trainings B or C and were found to be statistically significantly different as well. Repeating the analysis done in Table 6.6 but excluding all results from training A, results between evolved and fixed agent appear more significantly different as shown in Table 6.7.

**Table 6.7: Statistical analysis of evolved versus fixed agents without Training A results**

<table>
<thead>
<tr>
<th></th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test D</td>
<td>0.2207</td>
</tr>
<tr>
<td>Test E</td>
<td>0.0222</td>
</tr>
<tr>
<td>Test F</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

On the whole it could be argued that employing one of the above fixed strategies without any learning does not appear very successful and produces varied results.
Based on these results, it appears evident that evolved agents perform as well or better than fixed agents as a group, and therefore offer potential following further development and refinement.

6.2.3 Mutation, Immigration and changes to the Elite and Removed Populations

As an extension to Section 6.2.1 and to implement the refinements just suggested in Section 6.2.2, possible benefits of mutation versus immigration are investigated as well as exploring potential optimization of the elite and removed population sizes in the overall agent population. In the previous experiments, a significant lack of conversion in evolved populations could be observed based on a wide spread of results in training and testing. It is not clear whether increased conversion could improve results, as it might equally have an adverse effect by over specializing the agents for that particular data set, effectively decreasing their ability to perform in out-of-sample testing. These experiments are aimed at clarifying this through altering the elite and removed populations. Furthermore, mutation is compared to immigration as a tool to facilitate the continuous introduction of new genetic material and the potential acceleration of conversion. Their impact is compared using the agent’s performance as with the variations in the elite versus removed population sizes.
Experimentation and Results

A different data set to Section 6.2.1 was used, covering the time range from 01.01.1980 to 31.12.2002. To evolve the populations of 100 agents each, three training data sets were again created. Set G ranged from 01.01.1980 to 28.12.1984, set H from 01.01.1985 to 29.12.1989 and set I from 01.01.1990 to 30.12.1994. As shown earlier in Section 6.2.1, these time periods were assumed to be long enough to evolve a competitive population. The out-of-sample testing period covered 01.01.1997 to 31.12.2002, referred to as set J in Section 3.3. The fitness function used the total area under their total asset graph over the entire training period, namely \( FT_1 \).

Five populations were evolved on each data set for every one of the five parameter setups shown in Table 6.8, resulting in 75 results. After exposing the elite from each population to the out-of-sample data, the average from each setup was used for further comparison. The parameters used and averaged results obtained when investigating the elite to removed population sizes are shown in Table 6.8.

Results were compared statistically using a Kruskal-Wallis test, to assess whether results were consistent across data sets and whether a change in the elite to removed population sizes affected performance, as seen in Tables 6.9, 6.10 and 6.11.

As is clearly evident from Tables 6.10 and 6.11, there exists no significant difference when varying the elite to removed population sizes in the simulations performed. Further, results are predominantly consistent across data sets as demonstrated by Table 6.9, with the sole exception being A3, B3 and C3 showing a significant difference...
Table 6.8: Parameters used and corresponding results

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
<th>Elite</th>
<th>Removed</th>
<th>Performance (x10^8)</th>
<th>Std. Dev. (x10^8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>G</td>
<td>25</td>
<td>25</td>
<td>2.80</td>
<td>0.97</td>
</tr>
<tr>
<td>A2</td>
<td>G</td>
<td>10</td>
<td>25</td>
<td>2.94</td>
<td>1.62</td>
</tr>
<tr>
<td>A3</td>
<td>G</td>
<td>5</td>
<td>25</td>
<td>2.90</td>
<td>0.55</td>
</tr>
<tr>
<td>A4</td>
<td>G</td>
<td>25</td>
<td>10</td>
<td>2.38</td>
<td>0.32</td>
</tr>
<tr>
<td>A5</td>
<td>G</td>
<td>25</td>
<td>5</td>
<td>3.38</td>
<td>1.16</td>
</tr>
<tr>
<td>B1</td>
<td>H</td>
<td>25</td>
<td>25</td>
<td>3.66</td>
<td>0.71</td>
</tr>
<tr>
<td>B2</td>
<td>H</td>
<td>10</td>
<td>25</td>
<td>3.74</td>
<td>1.38</td>
</tr>
<tr>
<td>B3</td>
<td>H</td>
<td>5</td>
<td>25</td>
<td>3.68</td>
<td>0.76</td>
</tr>
<tr>
<td>B4</td>
<td>H</td>
<td>25</td>
<td>10</td>
<td>3.65</td>
<td>1.23</td>
</tr>
<tr>
<td>B5</td>
<td>H</td>
<td>25</td>
<td>5</td>
<td>3.71</td>
<td>0.63</td>
</tr>
<tr>
<td>C1</td>
<td>I</td>
<td>25</td>
<td>25</td>
<td>2.91</td>
<td>0.39</td>
</tr>
<tr>
<td>C2</td>
<td>I</td>
<td>10</td>
<td>25</td>
<td>2.98</td>
<td>1.42</td>
</tr>
<tr>
<td>C3</td>
<td>I</td>
<td>5</td>
<td>25</td>
<td>2.28</td>
<td>0.23</td>
</tr>
<tr>
<td>C4</td>
<td>I</td>
<td>25</td>
<td>10</td>
<td>2.41</td>
<td>0.28</td>
</tr>
<tr>
<td>C5</td>
<td>I</td>
<td>25</td>
<td>5</td>
<td>2.44</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table 6.9: Comparison by data set

<table>
<thead>
<tr>
<th>Experiment IDs compared</th>
<th>H-value</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 B1</td>
<td>5.18</td>
<td>0.0750</td>
</tr>
<tr>
<td>A2 B2</td>
<td>0.96</td>
<td>0.6188</td>
</tr>
<tr>
<td>A3 B3</td>
<td>8.06</td>
<td>0.0178</td>
</tr>
<tr>
<td>A4 B4</td>
<td>4.38</td>
<td>0.1119</td>
</tr>
<tr>
<td>A5 B5</td>
<td>5.78</td>
<td>0.0556</td>
</tr>
</tbody>
</table>

Table 6.10: Comparison by elite

<table>
<thead>
<tr>
<th>Experiment IDs compared</th>
<th>H-value</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 A2</td>
<td>0.78</td>
<td>0.6771</td>
</tr>
<tr>
<td>B1 B2</td>
<td>0.06</td>
<td>0.9704</td>
</tr>
<tr>
<td>C1 C2</td>
<td>3.14</td>
<td>0.2080</td>
</tr>
</tbody>
</table>

in results.

Following this, altering immigration and mutation ratios as well as changes to
6. AN INVESTIGATION INTO STATIC AND ADAPTIVE SYSTEMS

Table 6.11: Comparison by removed

<table>
<thead>
<tr>
<th>Experiment IDs compared</th>
<th>H-value</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 A4 A5</td>
<td>2.00</td>
<td>0.3679</td>
</tr>
<tr>
<td>B1 B4 B5</td>
<td>0.06</td>
<td>0.9704</td>
</tr>
<tr>
<td>C1 C4 C5</td>
<td>3.26</td>
<td>0.1959</td>
</tr>
</tbody>
</table>

the mutation rate were investigated. The experiment setup and results were as in Table 6.12, with elite and removed each set at 25.

Table 6.12: Parameters used and corresponding results

<table>
<thead>
<tr>
<th>ID</th>
<th>Data</th>
<th>Immigration to Mutation rate</th>
<th>Mutation Rate</th>
<th>Performance (x10^8)</th>
<th>Std. Dev. (x10^8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A6</td>
<td>G</td>
<td>1:1</td>
<td>0.2</td>
<td>3.20</td>
<td>1.00</td>
</tr>
<tr>
<td>A7</td>
<td>G</td>
<td>3:1</td>
<td>0.2</td>
<td>3.33</td>
<td>0.80</td>
</tr>
<tr>
<td>A8</td>
<td>G</td>
<td>1:3</td>
<td>0.2</td>
<td>2.47</td>
<td>0.48</td>
</tr>
<tr>
<td>A9</td>
<td>G</td>
<td>1:1</td>
<td>0.1</td>
<td>2.61</td>
<td>0.42</td>
</tr>
<tr>
<td>A10</td>
<td>G</td>
<td>1:1</td>
<td>0.4</td>
<td>3.24</td>
<td>1.46</td>
</tr>
<tr>
<td>B6</td>
<td>H</td>
<td>1:1</td>
<td>0.2</td>
<td>2.75</td>
<td>0.64</td>
</tr>
<tr>
<td>B7</td>
<td>H</td>
<td>3:1</td>
<td>0.2</td>
<td>3.34</td>
<td>1.41</td>
</tr>
<tr>
<td>B8</td>
<td>H</td>
<td>1:3</td>
<td>0.2</td>
<td>3.86</td>
<td>0.65</td>
</tr>
<tr>
<td>B9</td>
<td>H</td>
<td>1:1</td>
<td>0.1</td>
<td>3.42</td>
<td>1.98</td>
</tr>
<tr>
<td>B10</td>
<td>H</td>
<td>1:1</td>
<td>0.4</td>
<td>3.46</td>
<td>1.10</td>
</tr>
<tr>
<td>C6</td>
<td>I</td>
<td>1:1</td>
<td>0.2</td>
<td>2.77</td>
<td>1.57</td>
</tr>
<tr>
<td>C7</td>
<td>I</td>
<td>3:1</td>
<td>0.2</td>
<td>2.09</td>
<td>0.47</td>
</tr>
<tr>
<td>C8</td>
<td>I</td>
<td>1:3</td>
<td>0.2</td>
<td>2.61</td>
<td>0.98</td>
</tr>
<tr>
<td>C9</td>
<td>I</td>
<td>1:1</td>
<td>0.1</td>
<td>2.13</td>
<td>0.50</td>
</tr>
<tr>
<td>C10</td>
<td>I</td>
<td>1:1</td>
<td>0.4</td>
<td>2.63</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Using the Kruskal-Wallis test again, the following results were obtained as shown in Tables 6.13, 6.14 and 6.15.

In reference to Table 6.13, no clear indication exists as to whether the immigration to mutation ratio has a significant impact on the agent population. Comparing
Table 6.13: Comparing immigration versus mutation

<table>
<thead>
<tr>
<th>Experiment IDs compared</th>
<th>H-value</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A6 A7 A8</td>
<td>4.819</td>
<td>0.0899</td>
</tr>
<tr>
<td>B6 B7 B8</td>
<td>6.772</td>
<td>0.0338</td>
</tr>
<tr>
<td>C6 C7 C8</td>
<td>2.292</td>
<td>0.3178</td>
</tr>
</tbody>
</table>

Table 6.14: Variations in mutation rate

<table>
<thead>
<tr>
<th>Experiment IDs compared</th>
<th>H-value</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A6 A9 A10</td>
<td>0.608</td>
<td>0.7378</td>
</tr>
<tr>
<td>B6 B9 B10</td>
<td>0.749</td>
<td>0.6878</td>
</tr>
<tr>
<td>C6 C9 C10</td>
<td>2.351</td>
<td>0.3087</td>
</tr>
</tbody>
</table>

Table 6.15: Consistency across data sets

<table>
<thead>
<tr>
<th>Experiment IDs compared</th>
<th>H-value</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A6 B6 C6</td>
<td>0.082</td>
<td>0.9599</td>
</tr>
<tr>
<td>A7 B7 C7</td>
<td>10.749</td>
<td>0.0046</td>
</tr>
<tr>
<td>A8 B8 C8</td>
<td>8.433</td>
<td>0.0148</td>
</tr>
<tr>
<td>A9 B9 C9</td>
<td>2.351</td>
<td>0.3087</td>
</tr>
<tr>
<td>A10 B10 C10</td>
<td>1.836</td>
<td>0.3993</td>
</tr>
</tbody>
</table>

experiments A6, A7 and A8 as well as C6, C7 and C8 no significant difference can be found, while when comparing B6, B7 and B8 a significant difference does exist at a significance value of 0.05. This therefore suggests that both immigration and mutation have a similar effect on the evolutionary process and the immigration to mutation ratio is negligible.

Considering Table 6.14, changing the rate of mutation clearly has no significant effect in the system.

In terms of consistency of results across data sets, as shown in Table 6.15, it is apparent that no significant difference exists in the experiments assessing the varia-
tions in the mutation rate. However, experiments A7, B7 and C7 as well as A8, B8 and C8 which varied the immigration to mutation rates show a significant difference of results across data sets, as would be expected given their results in Table 6.13.

6.3 Adaptive System

There exist a variety of arguments in favor of adopting an adaptive approach to evolving trading strategies. Leiviskä for instance argued that any system must be capable of indefinitely continuing its training to enable adaptation, and must not be frozen after an initial training period [71]. To accomplish this, he argued that a system must be able to detect changes in the system, to which it can then respond by changing itself and therefore adapting to the change. Previous work on this by Trojanowski and Michalewicz also suggested that change could be initiated through deduction, where a change in the environment is noted by a fall in performance [130]. The obvious advantage offered by such a system would be its ability to continue operations fairly autonomously, without requiring intervention by a user, whenever a change in the environment is perceived. Furthermore, following this argument, sub-optimal performance is an inevitable disadvantage of any static system, as they lack the ability to continuously re-optimise its own parameters and thus eventually representing an obsolete solution [24].

Angelov et al extended this argument further by suggesting the need for smart adaptive systems [6]. These, they propose, ought to possess the properties of evolving
an adaptation mechanism, accumulating experience, autonomy, intelligence and the ability to deal with unexpected input in addition to being able to evolve their own structure. Obviously, these capabilities are highly desirable and offer some guidance for further research in the field, as at the moment there remains a gap in achieving these aims. In other application areas, adaptive approaches have already shown themselves successful, suggesting the need for wider implementation [11, 54, 65].

Pi and Rögnvaldsson provide a good example where an adaptive approach was found to be beneficial [96]. In their paper presenting a futures trading system using a neural network, they retrained their model with every additional piece of data that became available, which they found helped them overcome the non-stationary problem in financial data series as well as made more efficient use of available historical data.

6.3.1 Evolutionary Algorithm

The adaptive approach is defined by its continuous retraining of the agent population. Consequently, the data is not divided into training and testing data but represents one continuous set. The agent population was retrained after every trading day for 50 generations using the most recent 50 trading days closing price information based on previous exploratory work.

To achieve this, a trading population (TP) is first established using randomly generated genomes. In this experiment, the TP size was 25 agents, equivalent to that of the elite in earlier experiments for later performance comparison. For the
initial training phase and for every retraining phase during the experiment, a second population is created referred to as the evolution population (EP). The EP has a population size of 100, with a replica of the TP making up 25% of the EP, a further 25% are new randomly generated agents and the remaining 50% are offspring from the TP using two-point crossover. The EP then evolves using the same genetic algorithm, as outlined in Section 3.4.2 and, once finished, updates the TP and is then discarded. It is important to emphasize however that the TP does not necessarily change from trading day to trading day, but less successful agents in the TP have their genome replaced by the genome of more successful agents that have emerged from the EP. This way holdings and capital remain continuous in the TP. This could be loosely based on an analogy where a group of traders who each manage their own individual portfolio in a company, visit a course for a week, then later return to their role with a possibly new perspective on how to trade but with the same holdings and capital position. Once the TP is updated, the EP is removed and only recreated at the end of the next trading day to restart the whole process. A representation of this is shown in Figure 6.2.

Fitness was determined by comparing the total area under their total asset graph over the retraining duration.
6.3.2 Data

Data used in the system covers the time from 01.01.1996 to 31.12.2002. It is also important to note that 100 days of historical data prior to the first trading day are required to perform technical analysis. This is due to some indicators using large amounts of historical data in their calculations. For this reason a year of data preceding the test data set was kept for training, earlier referred to as data set L, to allow performance monitored trading to commence on the 1st of the year. Test data therefore covered 1565 trading days representing the time period 01.01.1997 to 31.12.2002, earlier referred to as data set J. This period was chosen as it exhibited significant movement, including the boom and crash periods around the millennium, but overall only resulted in the Index gaining 0.14%. It therefore considers a good range of trading environments within the trading period while overall essentially representing little change, allowing for effective comparison.
6.3.3 Fundamental Parameter Analysis

The impact of varying the quantity of data and the number of generations run for every re-training step in an adaptive trading system are investigated. Comparison is performed using three test sets, for which each has one variable altered for every run. The agent population was retrained after every trading day for Y generations using the most recent X trading days closing price information. Y and X representing the value that is used in every experiment, with experiments labelled A to G, and each experiment repeated five times and the average and standard deviation of those results shown in Table 6.16. For convenience, results from experiment A were repeated in the table.

<table>
<thead>
<tr>
<th>Exp. ID</th>
<th>Gen.</th>
<th>Days</th>
<th>Trading Profit (%)</th>
<th>Mean (S.D.)</th>
<th>Performance (x10^8)</th>
<th>Mean (S.D.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>50</td>
<td>50</td>
<td>95.10 (38.79)</td>
<td>2.84 (17.70)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>250</td>
<td>111.07 (150.62)</td>
<td>3.00 (1.251)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>250</td>
<td>10</td>
<td>48.10 (9.36)</td>
<td>2.61 (0.820)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>50</td>
<td>50</td>
<td>95.10 (38.79)</td>
<td>2.84 (17.70)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>50</td>
<td>25</td>
<td>12.25 (27.87)</td>
<td>2.30 (0.191)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>50</td>
<td>100</td>
<td>21.74 (5.11)</td>
<td>2.63 (0.251)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>50</td>
<td>50</td>
<td>95.10 (38.79)</td>
<td>2.84 (0.177)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>25</td>
<td>50</td>
<td>62.15 (36.08)</td>
<td>2.77 (0.360)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>100</td>
<td>50</td>
<td>46.57 (36.60)</td>
<td>2.64 (0.242)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The large standard deviation in Experiment B was due to one result set having returned a profit of 377.07% and a performance of 5.19x10^8. When removed and averaged across only 4 results, profit would have been 44.57% with a standard deviation
of 27.69% and performance would have been $2.45 \times 10^8$ with a standard deviation of $0.280 \times 10^8$ for experiment B, far more inline with the other results observed. Though no explanation can be offered for the outlier, it is important to note the potential this approach may produce.

Comparison between experiments A-F-G aimed to determine whether changing the number of generations used in retraining affected performance with a constant number of trading days. Comparison between A-D-E assessed whether changing the number of trading days used in retraining affected performance with a constant number of generations used. Experiments A-B-C maintained a constant number of days across the training phase (for example, 50 trading days over 50 generations equals 2500 days or 250 trading days over 10 generations equals 2500 days), but varied in trading days and generations run. If no difference could be observed, this would indicate that the change in either variable was counterbalanced by the change in the other. Kruskal-Wallis tests were done on each of those three groups using FT1 for comparison. Results are shown in Table 6.17.

<table>
<thead>
<tr>
<th>Comparison</th>
<th>H-value</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-B-C</td>
<td>4.380</td>
<td>0.1119</td>
</tr>
<tr>
<td>A-D-E</td>
<td>8.060</td>
<td>0.0178</td>
</tr>
<tr>
<td>A-F-G</td>
<td>1.820</td>
<td>0.4025</td>
</tr>
</tbody>
</table>

It is evident that there exists no significant statistical difference between A-B-C, implying that the deciding factor in retraining is the total exposure to data, as
determined by the number of retraining generations multiplied by the number of retraining trading days, rather than either parameter independently. Similarly, for A-F-G it seems that the system adapts after only few generations of retraining, as there exists no significant difference between the experiments either. Lastly, A-D-E also showed no significant difference at 0.01, implying that a change in trading days without compensation through additional generations run will also not affect performance.

Based on these results, it appears best to optimise the number of trading days in re-training for an adaptive system or risk sub-optimal performance. This could for instance be achieved or regulated by another evolutionary mechanism in the system. On the other hand compared to A-B-C and as no statistical significance exists, it is unlikely that when maintaining a constant number of generations, trading days should have a noticeable effect on performance in A-D-E. In summary, results indicate that varying retraining generations, varying the number of retraining days or maintaining an equal total exposure to data has no statistically significant effect on performance. However, further experimentation would likely provide greater insight and allow for more conclusive observations.

6.4 Comparison of Static and Adaptive Agent Systems

The focus here lies in the use of two general approaches to evolving successful systems, broadly classified into static and adaptive, to determine whether either presents
significant advantages over the other.

As introduced earlier, Allen and Karjalainen [4] first developed a system that used a genetic algorithm to create composite trading rules on which to base trading decisions. Following on from this work, Becker and Seshadri [13] and Potvin et al [98] used genetic programming based approaches. Many other papers have been published using various approaches ranging from neural networks to genetic programming and genetic algorithms, for example [33, 53, 66, 87, 108]. However all featured a similar segmentation of their data that can be classified as training and testing sets. This approach is referred to here as static evolution.

However, there are multiple inherent problems with using such a static approach. For instance, Pictet et al described the risks of over fitting [97], while similarly, Cheng and Yeh argued that in continuously changing environments an evolved solution might quickly become obsolete during testing [24]. Hence, if the aim of an approach is to obtain the best possible performance, a clear need exists to allow agents to continue evolution indefinitely, as discussed by Leiviskä [71]. He argues that a system must not be frozen after a learning phase but must be capable of adaptation. He continues by discussing the means necessary to achieve this as well as outlining the types of adaptation that exist. However for the purpose of this investigation the work will only draw on the argument that systems must not only continue to evolve or adapt, but in doing so will also perform better than frozen or static equivalents. Again, Pi and Rögnvaldsson [96] for example continuously retuned their neural network model,
minimizing the initial training data period while obtaining a very large out-of sample test. Similarly, Lettau looked at the concept of bounded rationality in agents using a genetic algorithm [73]. Interestingly, though Lettau used a continuous learning approach, he found that agents were still not able to “take rare events into account”, implying that even an adaptive approach may not be able to cope with some unusual events.

Rather than mainly assess performance of the presented system compared to the market overall or other strategies such as buy-and-hold, this work focuses on comparing the suitability of an adaptive approach compared to the more common static approach in the evolution of trading strategies. Genetic algorithms have been used in many applications, such as equity trading, just as the comparison of static and dynamic agent learning has been well studied. This work combines these two to investigate whether adaptive learning is also useful in genetic algorithm based technical trading in a direct comparison, ceteris paribus.

6.4.1 Experiment Setup

To facilitate a fair comparison of performance, in addition to using the same data for out-of-sample testing, the identical system setup was used for both the static and adaptive implementation. However, a few modifications were required for each, as outlined below.

It is also important to note that both approaches again require 100 days of histor-
ical data prior to the first trading day to perform technical analysis. For this reason a gap year, data set L, was inserted between the statics training data and the test data, to allow performance monitored trading to commence on the 1st of the year. As in Section 6.3.2, 1565 trading days worth of data was used representing the time period 01.01.1997 to 31.12.2002, earlier referred to as data set J.

Static Approach

Based on the same principle as the earlier experiments in Sections 6.2.1, 6.2.2, 6.2.3, six years worth of data prior to the gap year plus the testing period was used to train a population of 100 agents. 1565 trading days worth of data was used representing the time period from 01.01.1990 to 29.12.1995, referred to as data set K. As shown in Section 6.2.1 by H2 and H3, this time period was assumed to be long enough to evolve a competitive population. This population was trained using the genetic algorithm outlined in Section 3.4.2. The fitness function used the total area under their total asset graph over the entire training duration.

Adaptive Approach

Following from Section 6.3, two initialization procedures were used to also investigate their impact on results. For the first one, all 163 days of data available from the gap year were reserved for the initial training phase, run over 1000 generations of agents. In the other, the trained-up population from the static system of the corresponding
test run was used for initialization. The first set is referred to as i, while the latter as ii. In the first initialization process the TP is first established using randomly generated genomes, while for the second initialization process the elite of the evolved agents from the static approach were used to initialize the TP. By comparing results from both initialization methods it is possible to determine if there is any benefit in using a population trained over a longer time span compared to one using a short training span, which is likely to be very specific to the current environment as addressed by H1 in Section 6.2.1.

6.4.2 Results

For comparison, performance of both approaches over the test data was used, with performance of the buy-and-hold strategy shown to provide some contextual reference on how the market did overall. Each experiment was repeated five times with the average and standard deviation shown. Furthermore, results are shown as the profit obtained at the end of the period as well as its performance. Performance being defined as the area under their total asset graph over time, as in previous experiments.

Experiments are denoted as “S-” followed by a number if from the static system, or “A-” if from the adaptive system. Furthermore, A-1 to A-5 were initialized randomly and are collectively referred to as i, while A-6 to A-10 were initialized each using the evolved populations from S-1 to S-5 respectively and are collectively referred to as ii. Table 6.18 presents detailed results for each group, with Table 6.19 summarizing
their mean values for referencing convenience.

Table 6.18: Static and adaptive system results

<table>
<thead>
<tr>
<th></th>
<th>Profit (%)</th>
<th>Performance (x10^8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-1</td>
<td>23.74</td>
<td>2.51</td>
</tr>
<tr>
<td>S-2</td>
<td>26.19</td>
<td>2.47</td>
</tr>
<tr>
<td>S-3</td>
<td>11.42</td>
<td>2.38</td>
</tr>
<tr>
<td>S-4</td>
<td>33.65</td>
<td>2.62</td>
</tr>
<tr>
<td>S-5</td>
<td>07.90</td>
<td>2.47</td>
</tr>
<tr>
<td>Average</td>
<td>20.58</td>
<td>2.49</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.10691</td>
<td>0.09</td>
</tr>
<tr>
<td>A-1</td>
<td>120.68</td>
<td>2.79</td>
</tr>
<tr>
<td>A-2</td>
<td>51.26</td>
<td>2.78</td>
</tr>
<tr>
<td>A-3</td>
<td>87.29</td>
<td>2.86</td>
</tr>
<tr>
<td>A-4</td>
<td>69.12</td>
<td>2.64</td>
</tr>
<tr>
<td>A-5</td>
<td>147.13</td>
<td>3.12</td>
</tr>
<tr>
<td>Average</td>
<td>95.10</td>
<td>2.84</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.38791</td>
<td>0.18</td>
</tr>
<tr>
<td>A-6</td>
<td>54.51</td>
<td>2.82</td>
</tr>
<tr>
<td>A-7</td>
<td>43.51</td>
<td>2.66</td>
</tr>
<tr>
<td>A-8</td>
<td>141.92</td>
<td>3.83</td>
</tr>
<tr>
<td>A-9</td>
<td>90.59</td>
<td>3.59</td>
</tr>
<tr>
<td>A-10</td>
<td>31.35</td>
<td>2.94</td>
</tr>
<tr>
<td>Average</td>
<td>72.37</td>
<td>3.17</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.44732</td>
<td>0.51</td>
</tr>
</tbody>
</table>

Table 6.19: Summary of results

<table>
<thead>
<tr>
<th></th>
<th>Profit (%)</th>
<th>Performance (x10^8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy-and-Hold</td>
<td>02.34</td>
<td>2.70</td>
</tr>
<tr>
<td>Static</td>
<td>20.58</td>
<td>2.49</td>
</tr>
<tr>
<td>Adaptive (i)</td>
<td>95.10</td>
<td>2.84</td>
</tr>
<tr>
<td>Adaptive (ii)</td>
<td>72.37</td>
<td>3.17</td>
</tr>
</tbody>
</table>

As is evident from the results, the adaptive approach clearly outperforms its static equivalent in both profit at the end of the test period and, more importantly,
performance. As might have been expected, as the time period covered included the extreme market boom and crash around the millennium, generally classified as an anomalous period in wider literature, the adaptive approach fared better than its static equivalent.

In terms of agent profit, both static and adaptive outperformed the buy-and-hold strategy, with the adaptive in particular successfully outperforming it with respect to performance as well. However the absence of transaction costs must be highlighted in this respect.

Average performance for each trading day from each of the five runs for the three groups relative to the buy-and-hold strategy is illustrated in Figure 6.3.
Figure 6.3: Average performance graphs
6.4.3 Analysis and Discussion

In all cases the adaptive approach outperformed the static approach. As is clearly evident, whether initialized with the same set of knowledge as the static approach or using a different starting set of agents, the static approach failed to attain similar positive results. Overall, this observation is in-line with expectations, as the market represents a continuously changing complex environment where underlying rules, though generally valid, may from time to time need to be adjusted and different strategies will need to be employed. As the adaptive system provides this ability, it can exploit the emergence of successful short-term trading strategies over the general long-term strategies the static system is dependent on.

Analysing the difference between adaptive and static approaches, trading activity on a daily basis can also be considered and observations are shown in Table 6.20.

<table>
<thead>
<tr>
<th></th>
<th>Avg. Buys</th>
<th>Avg. Sells</th>
<th>Correlation (Std. Dev.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>3.88</td>
<td>1.49</td>
<td>0.963 (0.016)</td>
</tr>
<tr>
<td>Adaptive i</td>
<td>1.60</td>
<td>0.81</td>
<td>0.588 (0.180)</td>
</tr>
<tr>
<td>Adaptive ii</td>
<td>1.54</td>
<td>0.79</td>
<td>0.718 (0.126)</td>
</tr>
</tbody>
</table>

Table 6.20 illustrates the average number of daily buy and sell actions taken by each agent population from each approach, as well as the average correlation between their performance and that of buy-and-hold. Between the static and adaptive results, a noticeable difference exists in the number of trades conducted for both purchases.
and sales, indicating a more active approach taken by agents in the static approach. Both adaptive approaches also have noticeably different correlation values compared to the static approach, as further highlighted by Figure 6.3, reflecting the reduced trend-following trading behavior exhibited. Many strategies essentially represent a buffered buy and hold strategy, exhibiting a trend-following behavior with a reduced magnitude in movements, with little or no benefit gained from engaging in the market. This further does not yet consider the inclusion of transaction costs and other related fees. Lower market correlation and a higher performance on the other hand indicate a fundamentally successful trading strategy that is able to capitalize on gains and avoid loss situations successfully. Agents from the adaptive approaches were clearly able to demonstrate this.

Lastly, a Mann-Whitney U Test was used to compare the static results to each adaptive result set, and the adaptive sets to each other. Additionally, trading activity was compared between each group. Results are presented in Table 6.21.

<table>
<thead>
<tr>
<th>Test</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static – Adaptive (i)</td>
<td>0.00794</td>
</tr>
<tr>
<td>Static – Adaptive (ii)</td>
<td>0.00794</td>
</tr>
<tr>
<td>Adaptive (i) – Adaptive (ii)</td>
<td>0.30952</td>
</tr>
<tr>
<td>Static Buy – Adaptive (i) Buy</td>
<td>0.00008</td>
</tr>
<tr>
<td>Static Buy – Adaptive (ii) Buy</td>
<td>0.00008</td>
</tr>
<tr>
<td>Adaptive (i) Buy – Adaptive (ii) Buy</td>
<td>0.39710</td>
</tr>
<tr>
<td>Static Sell – Adaptive (i) Sell</td>
<td>0.00007</td>
</tr>
<tr>
<td>Static Sell – Adaptive (ii) Sell</td>
<td>0.00008</td>
</tr>
<tr>
<td>Adaptive (i) Sell – Adaptive (ii) Sell</td>
<td>0.10995</td>
</tr>
</tbody>
</table>
6. AN INVESTIGATION INTO STATIC AND ADAPTIVE SYSTEMS

There exists a significant difference between the results obtained from the static and adaptive approach, supporting the above conclusions that the adaptive approach outperforms the static. Furthermore, the remaining results indicate that a significant difference exists between static and adaptive trading behavior, with no significant difference observed between the two adaptive groups.

Based on these results, it appears that an adaptive approach offers clear advantages over its static alternative for optimizing trading strategies or populations using technical analysis. Though both implementations used here could be optimized further, by fine tuning re-training periods or population sizes for instance, neither was really given this advantage. Furthermore, the previous results seemed to indicate a negligible impact of varying those factors. Overall therefore it seems advisable to use an adaptive approach in future research attempting to optimize trading strategies using technical trading rules.

6.4.4 Agent Performance versus Benchmarks

With increasing processing power and ever more sophisticated prediction models, automated trading systems and strategies for use in financial markets are becoming increasingly popular. However, the ability of automated trading systems to outperform their human equivalent portfolio managers, the market itself or other financial alternatives, such as fixed interest bonds, is still debated. To add context to the performance of the agents, comparison to benchmarks such as the Index, a buy-and-hold
strategy and leading funds would provide a better understanding of how the agents actually performed.

Table 6.22 compares the agents average performance to a number of leading funds as specified by name and International Securities Identification Numbers, the buy and hold strategy and a fixed interest return of 5%.

<table>
<thead>
<tr>
<th>Description</th>
<th>Performance (x10^8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 6.18 Static Mean</td>
<td>2.49</td>
</tr>
<tr>
<td>Table 6.18 Adaptive (i) Mean</td>
<td>2.84</td>
</tr>
<tr>
<td>Table 6.18 Adaptive (ii) Mean</td>
<td>3.17</td>
</tr>
<tr>
<td>Buy-and-hold strategy</td>
<td>2.701</td>
</tr>
<tr>
<td>5% Fixed Interest</td>
<td>1.162</td>
</tr>
<tr>
<td>OP DAX (DE0008486382)</td>
<td>2.678</td>
</tr>
<tr>
<td>DekaFonds (DE0008474503)</td>
<td>2.628</td>
</tr>
</tbody>
</table>

The buy-and-hold strategy and both sample funds used display similar performance, with the fixed interest strategy faring worst. The evolved agents from Table 6.18 using the static approach performed slightly worse than the funds or the buy-and-hold strategy. The agents from Table 6.18 using the adaptive approach for both initialization methods however displayed a noticeably higher performance than the other investments, suggesting they outperformed the market, professionally managed funds as well as the buy-and-hold strategy, confirming the significant potential the approach holds.
6.5 Conclusion

This chapter presented an overview of intelligent agents trading on historical equity market data using technical indicators. It introduced a static and adaptive approach to evolving agents as well as optimizing and contrasting their performance among themselves and a number of benchmarks.

Optimization of the static system’s parameters were investigated in depth, including the effects on performance of variations in training data, population sizes, evolutionary cycles and the number of securities used. Experiments demonstrated that agents trained on five years worth of data performed significantly better than agents trained on only two years, while they did not show any statistical difference to agents trained on ten years worth of data across numerous experimental set-ups. Variations in population sizes, evolutionary cycles and number of securities included in the simulation where further studied. No significant differences were observed, except for a consistently lower performance for the largest agent population implemented. The parameters of elite to removed ratios were also investigated but findings showed that they did not affect performance. The introduction of mutation and varying rates of mutation to immigration suggested that either method will produce similar results. The evolved agents also outperformed agents using a fixed genome, the latter making decisions based on standard trading rules taken from literature.

Similarly, the adaptive system was also optimized, investigating a variation in the number of generations and the number of trading days used for re-training. The
results indicated that varying any of the variables in the range covered by this experiment did not significantly affect results. Comparing performance between the static and adaptive systems, the adaptive proved clearly superior. When compared to popular benchmarks, the adaptive approach clearly outperformed the Index, fixed interest and fund alternatives while the static approach on the other hand only fared comparatively to the benchmarks. Overall the experiments have demonstrated that using evolved agent populations for trading decisions can produce comparatively good performance. The comparison of static and adaptive approaches further highlights the scope of improvements possible in the portfolio optimization domain which can significantly improve results.

Following on from these findings, there are several further aspects that could yield improvements. More specifically, the potential of amalgamating decisions of individual agent populations into overall decisions and strategies, whether benefits exist in the implementation of teams, as well as if methodologically heterogeneous population could further offer any advantages. Some of these issues have been addressed in Chapter 7, where though the decision process required to amalgamate the information and decisions produced by agent populations is likely to require further study, the merit of hybridization on a basic methodological level is clearly evident and would again suggest potential for greater exploitation.
6.6 Summary

This section focused on an in-depth comparison of a static and adaptive system.

First, the static system was analyzed using a variety of system settings to determine their impact on the agent population’s performance. Results there showed that agents require a minimum amount of training data prior to being able to evolve fairly robust trading strategies that will produce competitive results across testing data. Equally however the improvement in results diminished as training data increased until no significant benefits could be observed from increasing the amount of training data. Following on from this a fundamental comparison was made between using the genetic algorithm to evolve and agents using a fixed genome based on values approximating those suggested in wider literature. Results showed that evolving agents does improve or at least equal the performance of fixed agents, justifying the increased computational expense of such an approach. The last part investigating the static system settings was a foray into the levels of mutation to immigration and elite to removed populations, increasing and decreasing either relative to each other. Results remained vague suggesting that altering ratios within the bounds of this experiment have no significant effect on agent performance, as well as demonstrating that mutation and immigration have a comparable effect on the agents, representing an arguably interchangeable element in the evolutionary process.

The adaptive system was then investigated independently from the static, introducing its key differences to the static, following which the impact of varying trading
days generations used at every evolutionary cycle was studied. Findings suggested that the number of trading days used in re-training the agent population does play an important role as, relating this back to the findings in Section 6.2.1, the agent performance is directly dependent on the available training data and noticeably more sensitive as the amount of training data diminishes. Though this can be counter balanced increasing the number of generations used, it seems that a competitive solution was found quickly based on the settings used in this series of experiments.

Lastly, the adaptive and static systems were compared directly. Both were tested on the same out-of-sample data, with the static system being trained on sufficient data to create a competitive solution in advance. The adaptive system similarly evolved an initial population, but as an extension to the experiment, was also initialized by the solutions evolved from the static system. This was done to allow for a direct comparison and see whether the adaptive re-training of those solutions would achieve a higher performance. Both systems were again benchmarked against the buy-and-hold strategy to provide additional context to their performance evaluation. Overall the static system was clearly outperformed by the adaptive system for both initialization methods, suggesting the latter being the superior choice in this system.
Chapter 7

An Investigation into Centralized Decision Making

7.1 Introduction

In this chapter the use of centralized decision makers is compared to groups of individuals in a portfolio optimisation simulation, investigating whether either offers benefits over the other. Though the merit of groups of independently trading agents in a variety of market settings has been extensively studied [4, 134, 64, 126], the potential benefits offered through group synergies needs to be explored further in this domain.

This chapter will investigate whether any benefits can be observed from a centralized decision making process, whereby an overall trading decision is derived from the decisions made by a population of technical traders.
7. Central decision maker agent

Prior to being able to design a centralized decision system, the issue of deriving a single sequence of decisions from a population of traders needs to be addressed. Currently, each agent in the population makes its own decisions and holds its own portfolio. However, given various possible approaches or methods of amalgamating the population’s individual decisions into one, it is important to determine which of these amalgamation methods will actually result in performance similar to that demonstrated by the population overall. In other words, using one amalgamation approach may result in a sequence of decisions that do not perform very well, while another might equally well significantly outperform other strategies. The question addressed here will be whether an amalgamation of decisions from individual agents is feasible and result in comparable performance to that of the population average, as deriving a single decision from a pool of optimized experts may result in a less optimal solution. This is based on the hypothesis that given a population of agents each holding their own portfolio and making their own decisions, it is possible to derive an amalgamation of these decisions that when applied to the market would result in performance comparable to the population’s average.

Four amalgamation methods were implemented in this work, each based on the collection of buy, sell or hold decisions made on each security considered by all agents for every trading day. For example, agent 1 has security 8 as a buy, agent 2 has security 8 as a sell and agent 3 has security 8 as a buy again. This would give
7. An investigation into centralized decision making

security 8 a count of 2 buys and 1 sell for the amalgamation process. In other words, this process essentially represents a simple voting system.

i. The first method (CDM1) performs a simple comparison between the number of overall buy and sell decisions. If either is greater than the other, the corresponding action is adopted for that security.

ii. The second method (CDM2) generates a buy or sell decision if at least half the population generated a buy or sell decision respectively.

iii. The third method (CDM3) will generate a buy or sell decision if at least one agent from the population generates the respective decision and no opposing decisions exist for that security.

iv. The fourth method (CDM4) follows the same principle as CDM3, except at least half the population must have generated the same decision for it to generate the corresponding decision.

An overview of the decision process from technical analysis to the central decision making agents is shown in Figure 7.1.

7.3 Experimentation and results

Data used in the system covers the time range from 01.01.1990 to 31.12.2002. To evolve the population of 100 agents, data set K was used with 1565 trading days
Central Decision Makers (CDM)

<table>
<thead>
<tr>
<th>CDM1:</th>
</tr>
</thead>
<tbody>
<tr>
<td># Buys &gt; # Sells -&gt; Buy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CDM2:</th>
</tr>
</thead>
<tbody>
<tr>
<td># Buys &gt; # Sells &amp; # Buys &gt;= 25/2 -&gt; Buy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CDM3:</th>
</tr>
</thead>
<tbody>
<tr>
<td># Buys &gt; 0 &amp; # Sells = 0 -&gt; Buy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CDM4:</th>
</tr>
</thead>
<tbody>
<tr>
<td># Buys &gt; 0 &amp; # Sells = 0 &amp; # Buys &gt;= 13 -&gt; Buy</td>
</tr>
</tbody>
</table>

Buy/sell decisions for all 25 agents are passed to CDMs

Agent 1, Agent 2, ... Agent 25

<table>
<thead>
<tr>
<th>DT1:</th>
</tr>
</thead>
<tbody>
<tr>
<td># Buys &gt; # Sells -&gt; Buy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DT2:</th>
</tr>
</thead>
<tbody>
<tr>
<td># Buys &gt; # Sells &amp; # Buys &gt;= 8/2 -&gt; Buy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DT3:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum \text{signals} \times \text{weight} &gt; 0 ) -&gt; Buy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DT4:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum \text{signals} \times \text{weight} &gt; 0 ) &amp; ( \sum \text{signals} \times \text{weight} 10 ) -&gt; Buy</td>
</tr>
</tbody>
</table>

8 buy/sell/hold signals per security

<table>
<thead>
<tr>
<th>Simple Moving-average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Strength Index</td>
</tr>
<tr>
<td>Price-Rate-of-Change (short-term)</td>
</tr>
<tr>
<td>Price-Rate-of-Change (long-term)</td>
</tr>
<tr>
<td>Stochastic Oscillator (interpretation 1)</td>
</tr>
<tr>
<td>Stochastic Oscillator (interpretation 2)</td>
</tr>
<tr>
<td>Moving Average Convergence Divergence</td>
</tr>
<tr>
<td>Bollinger Bands</td>
</tr>
</tbody>
</table>

Figure 7.1: Decision process overview

worth of data, representing six year from 01.01.1990 to 29.12.1995. Again, as demonstrated in Section 6.2.1, this time period should be sufficient to evolve a competitive population. The out-of-sample period used data set J and was taken over six year from 01.01.1997 to 31.12.2002. The fitness function used the total area under their
total asset graph over the entire training duration, referred to as FT1, as well as the oGA as the evolutionary mechanism.

For comparison, ten agent populations were evolved and tested over the out-of-sample period. The elite’s performance from each population, as defined in previous chapters, was recorded as a benchmark. Four central decision makers were implemented, each corresponding to the four decision making types described. The decisions made by the elite population on every trading day were used by the central decision makers to generate their own decisions for every experiment run. The following results were obtained, as shown in Table 7.1.

<table>
<thead>
<tr>
<th></th>
<th>CDM1</th>
<th>CDM2</th>
<th>CDM3</th>
<th>CDM4</th>
<th>Elite</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.25</td>
<td>2.54</td>
<td>2.30</td>
<td>2.30</td>
<td>2.90</td>
</tr>
<tr>
<td>2</td>
<td>2.42</td>
<td>2.25</td>
<td>2.14</td>
<td>2.14</td>
<td>3.03</td>
</tr>
<tr>
<td>3</td>
<td>2.40</td>
<td>2.71</td>
<td>3.14</td>
<td>3.14</td>
<td>3.06</td>
</tr>
<tr>
<td>4</td>
<td>2.08</td>
<td>2.15</td>
<td>2.00</td>
<td>2.01</td>
<td>3.01</td>
</tr>
<tr>
<td>5</td>
<td>2.54</td>
<td>2.09</td>
<td>2.07</td>
<td>2.07</td>
<td>3.02</td>
</tr>
<tr>
<td>6</td>
<td>2.51</td>
<td>2.77</td>
<td>1.92</td>
<td>1.92</td>
<td>3.01</td>
</tr>
<tr>
<td>7</td>
<td>2.32</td>
<td>2.26</td>
<td>2.21</td>
<td>2.22</td>
<td>3.18</td>
</tr>
<tr>
<td>8</td>
<td>2.50</td>
<td>2.51</td>
<td>2.75</td>
<td>2.75</td>
<td>3.14</td>
</tr>
<tr>
<td>9</td>
<td>2.31</td>
<td>2.08</td>
<td>2.90</td>
<td>2.90</td>
<td>3.00</td>
</tr>
<tr>
<td>10</td>
<td>2.62</td>
<td>2.30</td>
<td>2.37</td>
<td>2.35</td>
<td>3.21</td>
</tr>
<tr>
<td>Mean</td>
<td>2.40</td>
<td>2.37</td>
<td>2.38</td>
<td>2.38</td>
<td>3.06</td>
</tr>
<tr>
<td>Buy &amp; Hold</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.70</td>
</tr>
</tbody>
</table>

CDM1, CDM2, CDM3 and CDM4 all showed similar average performance while the elite demonstrated significantly better results.

Two sets of ten runs were performed, with the first providing identical starting
capital to each CDM as to every trader in the elite (100,000), as shown in Table 7.1, while the second provided each CDM with a starting capital equivalent to the sum of the elite’s starting capital (25*100,000=25,000,000). However, the latter experiment showed very similar results that reproduced the ones observed in Table 7.1. In conclusion, incrementing the starting capital at those levels does not affect trading performance and will therefore not be considered further here.

A graphical representation of the averaged performance of the elite and each CDM over the out-of-sample test period is shown in Figure 7.2, also including a comparison of their performance to the buy-and-hold strategy. Figure 7.2 shows that the elite outperformed the buy-and-hold strategy across the entire time span, while the CDMs underperformed for most of the out-of-sample data. The buy-and-hold strategy has a total area under its graph of 2.70x10^8, which is also clearly larger than that shown by the CDMs.
Figure 7.2: Average performance of CDMs
7.4 Analysis and discussion

Comparing the mean from the ten experiments for each CDM and the elite, it is quite apparent that the four amalgamation processes did not produce a phenotype that is as successful as the elite. Furthermore, when compared to the buy-and-hold strategy only the elite offers a successful alternative. Statistical analysis using Kruskal-Wallis compared CDM1 to CDM4 among one another to determine if there exists any significant difference between them. CDM1, CDM2, CDM3 and CDM4 were compared individually to the elite using Mann-Whitney U Tests to determine if a significant difference exists between the elite and the central decision makers. Results are shown in Tables 7.2 and 7.3 respectively.

**Table 7.2: Kruskal-Wallis test**

<table>
<thead>
<tr>
<th>Experiments compared</th>
<th>H-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM1, CDM2, CDM3, CDM4</td>
<td>1.181</td>
<td>0.7576</td>
</tr>
</tbody>
</table>

**Table 7.3: Mann-Whitney U tests**

<table>
<thead>
<tr>
<th>Experiment IDs compared</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elite CDM1</td>
<td>0.0002</td>
</tr>
<tr>
<td>Elite CDM2</td>
<td>0.0002</td>
</tr>
<tr>
<td>Elite CDM3</td>
<td>0.0019</td>
</tr>
<tr>
<td>Elite CDM4</td>
<td>0.0019</td>
</tr>
</tbody>
</table>

As is clearly apparent, no significant difference in performance exists between the amalgamation methods. However, results demonstrate that a significant difference in performance exists between each amalgamation method and the elite. Near identical...
results were observed for the experiments using the alternative starting capital for CDMs as well.

It could be argued that the CDMs did not attain comparable results to the elite based on their trend following behaviour relative to the elite’s performance at a reduced magnitude. As can be seen in Figure 7.2, they closely follow the movements of the elite’s performance, as would be expected given the simple voting mechanism employed. This may suggest that at best, performance will be equal to the elite’s performance, implying a homogeneous and highly converged elite population. As a potential extension to this experimentation series, a diversity encouraging algorithm such as the rsGAs or rtGAs could be employed, which may lead to significantly different results.

In conclusion, this investigation demonstrated that distilling a single or centralized decision from a group of individual decision makers does not necessarily improve or maintain a comparable level of performance. The emphasis here lies on the amalgamation process itself, as it is the key factor in determining success of the approach and needs to be specifically designed and refined for the application domain.

7.5 Summary and conclusions

This section compared the use of a centralized decision maker to groups of individual agents. Four implementations of a centralized decision maker were introduced, representing a sample number of methods for amalgamating the individual agents’
decisions. Results were compared and showed that, with no difference existing between the different approaches to deriving a single decision for the centralized decision makers, the elite population of individual traders performed significantly better than any of the centralized decision makers. Furthermore, the centralized decision makers failed to outperform the buy-and-hold benchmark. This clearly demonstrated that amalgamating the decisions, thereby arguably reducing the risk element, did not affect performance positively in this implementation.
Chapter 8

Conclusions

8.1 Contributions

A novel portfolio optimization system based on an evolved population of agents, representing equity market traders, was presented. A variety of studies were performed using this system to explore a range of recurring outstanding questions found in literature.

8.1.1 The Impact of Trading Methodologies on Agent Performance

Defined as a static system, multiple agent populations were trained independently over a certain amount of data prior to being exposed to a set of out-of-sample data. All other system settings were kept constant, however, in order to reduce the number of variable factors. Further, the Capital Asset Pricing Model, Technical Analysis and a hybrid approach of both methods was used by agents as their trading methodology. The results clearly demonstrated that the choice of an agent’s methodology has a
noticeable impact on the agent’s performance. Specifically, agents using the Capital Asset Pricing Model as their methodology demonstrated the least desirable performance, while agents using Technical Analysis as their methodology produced overall better results. The hybrid method, consisting of Technical Analysis coupled with the Capital Asset Pricing Model, performed best overall, clearly demonstrating that a hybridization of methods can improve results and might allow for greater performance increases if studied further.

8.1.2 Variations in Genetic Algorithm Implementations and Fitness Functions

A significant variety of genetic algorithms and similar evolutionary methods have been employed in related studies, following from extensive research in the field itself. In this study the impact of the genetic algorithm’s design as well as the choice of fitness function on the simulated trader’s performance was explored in-depth. Using a number of permutations of Holland-style algorithm implementations as well as a custom implementation of a genetic algorithm, the performance of agents evolved by each system were contrasted and analyzed. The study demonstrated that based on the features employed, negligible differences could be observed in performance, supporting the wide variety of algorithms employed without any particular design having emerged superior. However, in terms of finding the most optimal solution on the training data, the oGA clearly performed better than all other explored algorithms while the tGA displayed the most rapid convergence. These observations in turn highlight
the need for customization based on the application environment, as for instance the tGA would clearly be a preferred choice in an adaptive or resource constraint system compared to the others presented. A variety of fitness functions were also explored to determine their impact on trading populations. Findings showed that significant differences exist between some fitness functions, producing better performing solutions more consistently. In essence, considering both studies in combination, findings support the large number of different evolutionary strategies and related fitness measures employed in research with seemingly no general rules being applicable, short of evaluating every system implementation on a case by case basis.

8.1.3 Independent Optimisation of Static and Adaptive Approaches

An in-depth look was taken at the comparison of a static versus an adaptive system as well as the optimization of each respectively. First the impact on agents’ performance in a static system was studied using a variety of system settings. The results demonstrated that a minimum amount of training data is required prior to being able to evolve consistent trading strategies that will achieve competitive results in out-of-sample testing. Improvement in the results diminished however as training data was increased, up to the point where no significant increase in benefits could be observed anymore. An additional comparison between the performance of evolved agents and agents using a fixed genome was made. The results showed that evolved agents can improve or at least equal the performance of that of fixed agents justifying
the increased computational expense of such an approach. Lastly, a brief experiment studied the impact in varying the ratios of mutation to immigration and ratios of elite to removed populations used in the system. The results however showed that altering ratios as shown here had no significant effect on agent performance. Further it demonstrated that mutation and immigration have a comparable effect on the agent’s performance, representing an arguably interchangeable element in the evolutionary process.

Taking a similar approach to the static system, this was followed by an investigation into the impact of using a number of settings for trading days and for generations used at every evolutionary cycle within the adaptive system. Results showed that the number of trading days used in re-training the agent population is an important factor, suggesting that agent performance is directly dependent on the available training data and is increasingly more sensitive to diminishing amounts of training data. Though possible to counter balance by an increase in the number of generations used, it appeared that a competitive solution can be found quickly based on the settings used here.

8.1.4 Comparison of Static versus Adaptive Approaches

Finally, the adaptive and static systems were compared directly. The same out-of-sample data was used for testing for consistency, with the static system being trained in advance to evolve an initial population. The adaptive system also evolved
an initial population in advance, however as an extension was also initialized with the static system’s initial population to determine whether the re-training of those solutions during the adaptive system’s process would achieve higher performance. Overall the static system was clearly outperformed by the adaptive system using both initialization methods, suggesting the latter being the superior choice.

8.1.5 The Effect of Amalgamating Individual Trading Decisions

The final investigation looked at the use of a centralized decision maker to attain better performance compared to groups of individual agents. Four centralized decision maker types were investigated, representing a selection of possible approaches of methods for amalgamating the decisions of the individual agents. However, in the results no difference was found between the different centralized decision makers with the individual traders performing significantly better. Furthermore, the centralized decision makers failed to outperform the buy-and-hold benchmark. In conclusion, reducing the risk element by amalgamating decisions did not have a positive effect on results.

8.2 Limitations and Further Work

Due to the vast depth of theories ranging from analysing securities to genetic algorithm implementations, only a relatively small sample was covered in this work with its primary aim being to establish a foundation from which to explore each respective
area further.

For instance, this system could easily be applied to other financial instruments or encompass a wider choice for agents in constructing their portfolios. This may fundamentally affect performance as shifting between volatile and less volatile investments may allow agents to capitalize even in downward moving markets. Particularly if derivatives were to be included, an agent could create significant returns based on volume analysis for instance and with a fairly simple mechanism create completely new revenue sources. If this simulation was adapted for use in a trading simulation competition for example, this may be particularly appealing.

As discussed in Chapter 2, multiple authors have proposed more responsive evolutionary systems which adapt based on changes in the environment. To cite Cobb’s hypermutable state, where mutation increases when environmental stress increases for instance [27], this would provide a very interesting avenue of research. Work in this area would be of particular use to manage abnormalities or unusual events, such as the recent global market fluctuations between the 21st and 24th of January 2008. Significant changes in prices were observed due to a number of factors, such as the perceived looming risk of recession, possible mass selling as may have resulted through the Société Générale scandal or the lingering sub-prime effects. Regardless of the underlying factors, such anomalous market movements can be argued to be better handled by a system which incorporated an element of adaptivity not just in the adaptive approach sense as in this work, but also adapting the actual evolution-
ary system used. In theory this could potentially serve to further minimize risk by increasing responsiveness and the underlying adaptation of agents to the new, albeit arguably temporary, environment.

Leading on from this, a limitation of the work presented here was the relatively small excursion into artificial intelligence methods used and presented. As the focus remained on genetic algorithms and a small set of its permutations, there remains a vast array of fundamentally differing approaches that were to some extent introduced in the literature review earlier. Particularly in the adaptive system, considerable improvements could be made using heuristics specifically designed for a faster exploration of the search space while maintaining the same population defining characteristics as with the genetic algorithms used.

The work presented in Chapter 7 demonstrates an amalgamation process deriving a single trading decision based on the decisions made by a population of agents using Technical Analysis. The investigation showed that the agents using the amalgamated decision consistently demonstrated lower performance than the optimized agent populations, suggesting that averaging optimized solutions does not offer any benefits in this situation. Coupled with the work in Chapter 4, this could be extended by introducing an alternative method of analysing securities, which in conjunction with the Technical Analysis based agent population may in turn offer advantages to deriving a centralized decision. Future work could focus on expanding the pool of heterogeneous agents the amalgamation process may draw on and investigate the effects it may have
on the derived decisions and decision process overall.

Lastly, the data used exclusively in this series of experiments was daily closing price information for stocks listed in the DAX 30 Index. As often stated in literature [22], the behavior of markets or the development of trends and price changes is a continuously changing process. Particularly in light of the dramatic increase in electronic and especially automated trading systems in recent years, this will have changed the development of prices in response to announcements or news in general dramatically. An interesting avenue of research building on the work presented here would be to explore if apparently robust solutions derived and successfully applied over the time spans in this work would still be applicable in the current markets. Determining the rate of change in market dynamics itself could prove a valuable insight and prove a significant competitive edge when taken into account in designing systems, as fundamental assumptions as were made here could be affected.

8.3 Reflections on the Original Hypotheses

The work presented here provides an in-depth examination of some of the factors that affect portfolio optimization in an agent based system on a fundamental level. The work showed the significant impact relatively minor design choices may have on the resulting system and contrasts this to the relative lack of exploration thereof in related research. It also addressed some of the most commonly used approaches to designing systems in an attempt to further understanding as to which may prove
more suitable in portfolio optimization problems.

The novel system implementation as a simulation environment for evaluating an automated trading system as a portfolio optimization problem proved very versatile, efficient and effective across the series of experiments. Using an agent orientated approach rather than focusing on strategy evolution allowed for a comparison to human trading behavior and analysis methods.

The choice of trading methodology will affect agent performance, with one method likely comparatively superior. Large numbers of different trading methodologies and methods of analysis exist in popular use. In order to determine their impact and potential applications in a trading simulation, particularly when using agents emulating traders, a small sample using technical analysis were implemented and compared to each other. The hybrid method in particular produced the best results suggesting further work into developing a more diverse methodology base for trading agents should result in improved performance. Overall the investigation clearly highlighted that the choice of methodology has a significant impact on results obtained. This therefore clearly establishes the need for empirical testing for justification of any chosen methodology in future research.

Small variations in the design of the genetic algorithm, as well as its parameters, can have a significant impact on agent performance in a system. A large number of genetic algorithm permutations have been im-
implemented and studied, offering improved performance or introducing features that may provide better results. In this study a number of basic variations of the Holland style algorithm were implemented, including features such as fitness scaling to incorporate the naturally occurring variety of approaches to trading among humans. Results found no difference in agent performance per se, however differences were observed in terms of rate of convergence to optimal solutions and the consistency with which these could be reproduced across trading methodologies investigated earlier. Though not conclusive, findings did suggest some algorithm implementations would be preferable in resource constraint systems for instance.

The choice of fitness function and particularly when used simultaneously as the measure of performance, has a significant impact on results and consequently derived conclusions. Assessing the performance of agents or more generally, the results produced by the system, are a key factor in determining whether results provided are relevant or of interest particularly when set in competitive environments. The fitness functions implemented in this study aimed to reflect the diversity in possible interpretations of performance, and assessed the impact when used as the genetic algorithms fitness function. Results clearly showed that some fitness functions will be preferable and produce better results than others, highlighting the need for a more elaborate selection procedure prior to implementing fitness functions in future simulations. Crucially,
differences could be observed between systems for the same fitness function, further supporting the need for preliminary testing as to the choice of a fitness function measured using a representative measure such as FT1 before finalizing a system implementation.

**An adaptive system offers greater advantages over its static equivalent and will demonstrate better performance.** A vast variety of system implementations use either a static or adaptive approach as defined in this work, with no study having clearly compared both approaches in an otherwise controlled and unvaried system. The study here showed that following initial optimization of parameters in both systems, the adaptive proved clearly superior as suggested in other literature and based on existing work in other domains. In essence, in any portfolio optimization situation where performance maximization is the aim, the use of a static system should therefore be discouraged.

**Amalgamating multiple agents’ trading decisions into one, by means such as consensus voting for example, will merely reduce trading volume and not improve overall results.** In human teams it can be observed that a group of capable individuals working together will outperform themselves in an individual setting, while this can not be observed in groups where the initial skill set is lacking. In an attempt to determine whether such synergies can be observed in this system implementation, a set of central decision makers were compared. Performance however did not suggest any need for or benefit from
creating such specialized agents, arguably as the lack of diversity among the
agents in terms of methodologies prevented a more successful selection process.

8.4 Final Remarks

Many studies have been performed investigating a vast variety of aspects in financial markets. The foci have ranged from furthering the understanding of market dynamics, clarifying the motivation and behavior of market participants to simply maximising the return on investments using financial instruments to name just a few. Though a multitude of methodologies and approaches have been taken to developing and implementing the systems used in these studies, no definitive studies have been made of the various aspects involved. These have been partially addressed in this work, forming an incursion into an open and fair comparison of some of the key factors underlying these studies.

8.5 Dissemination


C. Schoreels and J.M. Garibaldi, “Comparative study of central decision makers versus groups of evolved agents trading in equity markets”, Proceedings of the 12th


References


REFERENCES

Computational Intelligence and Natural Computing, pages 26–30, September 2003.


REFERENCES


[74] J. K. P. Lipinski. Evolutionary building of stock trading experts in a real-time


Evolutionsstrategie : mit einer vergleichenden Einhrung in die Hill-Climbing-

tems. Technical Report ULCS-02-005, Department of Computer Science, Uni-
versity of Liverpool, 2002.


January 1966.


1994.

[116] O. Shehory and A. Sturm. Evaluation of modeling techniques for agent-based
systems. In Proceedings of 5th ACM International Conference on Autonomous

[117] A. Sherstov and P. Stone. Three automated stock-trading agents: A compar-
ative study. In P. Faratin and J. Rodriguez-Aguilar, editors, Agent Mediated


APPENDIX A

Glossary of Terms

The definitions used below are cited from the Oxford Concise Dictionary [94], unless indicated otherwise.

Agent (software agent:) An entity in a program that operates following a specific set of rules.

Artificial Intelligence (the study of) the capacity of machines to simulate intelligent human behavior, or the subfield of computer science concerned with the concepts and methods of symbolic inference by computer and symbolic knowledge representation for use in making inferences [99].

Autonomous Agent An agent that has a certain independence of external control [95].

Cardinality The number of elements in a set [99].
**Crossover** The main operator used in artificial evolution for generating offspring from two parents. It works through choosing an insertion point and exchanging the string on one side with the corresponding string of the other parent [99].

**Cycle** A series of events that are regularly repeated in the same order. In this context, it is one interval of trading day.

**Equities** See securities definition.

**Equity** The value of shares issued by a company.

**Equity market securities** An asset entitling the owner to a share in an institution publicly listed on one or more indices allowing for trade in that market.

**Financial analysis** Methods of analyzing financial data, such as technical, fundamental and behavioral analysis.

**Fitness** The value of the fitness function, which is an optimisation criterion, where individuals with a high fitness are used in reproduction [95].

**Fund** A financial institution that sells shares to individuals and invests in securities issued by other companies [99].

**Fund Manager** The individual responsible for making portfolio decisions for a mutual fund, pension fund, or insurance fund [135].

**Gene** A unit of heredity which is transferred from a parent to offspring and is held to determine some characteristic of the offspring.
Genetic algorithm  An evolutionary population based algorithm inspired by the Darwinian theory of natural selection and survival of the fittest.

Genome  Complete set of genetic material in an organism.

Genotype  The genetic constitution of an individual organism.

Heuristics  Search methods which avoid brute force and take a satisficing approach to finding a hopefully near optimal solution in exchange for a reduced search time.

Index (pl.: Indices)  (Share Index:) A financial index formed by selecting a number of shares of prominent companies traded on a stock exchange and comparing the daily average price of those shares with the average price on a stated date in the base year. A weighted average is usually taken (i.e. it takes into account the value of each share traded) [3].

Intelligent Agent  An agent which is coupled with some form of heuristic which gives it the ability to act autonomously in a directed fashion.

Portfolio  A range of investments held by a person or organisation.

Qualitative Information  This information represents items that are not generally expressed in numerical terms or in many cases, cannot be expressed in a form that would be compatible with a current computer system. For example human emotions, sentiments or generally opinions. In the financial context for instance
the value attributed to 100 tons of balsa wood can vary from one individual to another.

**Quantitative Information/Data** This information represents raw data such as the price of a security or the temperature of a glass of water. As such data is often meaningless without context, providing one attributes meaning and hence allows easier interpretation, converting quantitative data to quantitative information. Often, a collection of quantitative data can represent quantitative information just as well as a single piece of data, such as is the case for averages for example.

**Securities** A certificate attesting the ownership of stocks or bonds, or the right to ownership connected with trade. However, it can also be defined as a formal declaration that documents a fact of relevance to finance and investment; the holder has a right to receive interest or dividends [99].

**Stakeholders** A person who has an interest or concern in something. Or, one who has a share or an interest, as in an enterprise [1].

**Straight-through-processing** Straight through processing is the end to end automation of the trading process within and between both buy and sell side institutions, from the first capture of an order through to final settlement. It involves the seamless, electronic transfer of information to all parties involved in the trading cycle utilising standardised information flows, technologies and
infrastructures [103].

**Trading Day** Any day on which trading occurred. Typically Monday to Friday for a regular week.
APPENDIX B

Training Genome Convergence

B.1 Genome Convergence using Technical Analysis
Figure B.1: TI oGA Genome Correlation
Figure B.2: TI sGA Genome Correlation
Figure B.3: TI tGA Genome Correlation
Figure B.4: TI rsGA Genome Correlation
Figure B.5: TI rsGAs Genome Correlation
Figure B.6: TI rtGAfs Genome Correlation
B.2 Genome Convergence using Hybrid
Figure B.7: Hybrid oGA Genome Correlation
Figure B.8: Hybrid sGA Genome Correlation
Figure B.9: Hybrid tGA Genome Correlation
Figure B.10: Hybrid rsGA Genome Correlation
**Figure B.11: Hybrid rsGAs Genome Correlation**
Figure B.12: Hybrid rtGAs Genome Correlation
B.3 Genome Convergence using the CAPM
Figure B.13: CAPM oGA Genome Correlation
Figure B.14: CAPM sGA Genome Correlation
Figure B.15: CAPM tGA Genome Correlation
Figure B.16: CAPM rsGA Genome Correlation
Figure B.17: CAPM rsGAs Genome Correlation
Figure B.18: CAPM rtGAs Genome Correlation
B.4 Genome Convergence using Sharpe
Figure B.19: Sharpe oGA Genome Correlation
Figure B.20: Sharpe sGA Genome Correlation
Figure B.21: Sharpe tGA Genome Correlation
Figure B.22: Sharpe rsGA Genome Correlation
Figure B.23: Sharpe rsGAs Genome Correlation
Figure B.24: Sharpe rtGAs Genome Correlation